CLUBE

Março/84

N.º 18

Coordenadores: MARIA IRENE • ALEXANDRE SOUSA • J. MAGALHÃES • AV. BOAVISTA, 832-2.º T. • TELEF. 65127 • 4100 PORTO

NESTE NÚMERO

Edição: Clube Z80

Fotocomposição: Fotomecânica Mabreu/Porto

Impressão: Gráfica Firmeza/Porto

Tiragem: 500 exemplares, Março 1984

INTRODUÇÃO À LINGUAGEM MÁQUINA * ZX81/Spectrum

Autor: FERNANDO PRECES
Sacavém

(Continuação)

CAPÍTULO II

PARTE I — *Fundamentos de microprocessadores*

A aparição do microprocessador provocou uma mudança radical no projecto dos sistemas digitais. Segundo o processo tradicional, muitas vezes chamado de «lógica aleatória», os sistemas são projectados usando-se os blocos lógicos individuais (tais como *flip-flops*, *gates* e *contadores*), necessários a cada aplicação.
A utilização da lógica aleatória implica uma configuração diferente para cada aplicação, havendo muito poucas semelhanças entre os diferentes sistemas. É um processo de projecto identico ao dos circuitos analógicos — uma vez construído o circuito, é extremamente difícil adaptá-lo a qualquer novo tipo de funcionamento.
O microprocessador permite a construção dum sistema geral, que pode ser adaptado a uma larga variedade de aplicações sem ser necessário realizar grandes modificações. A individualidade dos diferentes sistemas existe na lista de instruções (chamada «programa») que controla o seu funcionamento. Há, pois, duas componentes nos sistemas com microprocessadores:

— Os circuitos (a que se chama o «hardware»)
— Os programas (o «software»).

Será feito um pequeno resumo sobre o hardware, para sabermos de que elementos é composto um sistema de microprocessador. Depois estudaremos programação e a forma de *unir as duas componentes.*

1.1 — O desenvolvimento do microprocessador

Os computadores electrónicos mais antigos eram construídos com válvulas, aos milhares. Eram máquinas enormes e com avarias constantes.
A segunda geração já utilizava transistores. Conseguia-se assim máquinas não só mais pequenas e mais fiáveis como também mais baratas.
Foi esta geração que marcou o início da era da expansão dos computadores como equipamentos de larga utilização.
Em 1960, computadores ainda mais pequenos e mais poderosos foram construídos à custa da utilização, pela primeira vez, de milhares de *"gates"*, *"flip-flops"* e outros elementos sobre a forma de *circuitos integrados SSI.*
O desenvolvimento da tecnologia dos semicondutores possibilitou, a breve trecho, a integração de dezenas de "gates" num único circuito.
A tendência para a miniaturização manteve-se e, em 1971, o primeiro microprocessador apareceu no mercado. Continha num único circuito integrado a quase totalidade das secções de control e cálculo ("Unidade de Processamento Central — CPU") de um computador.
Um microprocessador contém milhares de "gates", pertencendo por isso à categoria dos circuitos integrados em Larga Escala (LSI). Paralelamente a ele foram desenvolvidas memórias, também LSI, capazes de armazenar milhares de bits.

1.2. — Configuração básica de um microprocessador

Interessa-nos, em primeiro lugar, considerar um sistema central que possua dois periféricos:

— *O periférico de entrada* (Teclado), que permite a introdução de instruções e dados.
— *O periférico de saída*, aonde se visualiza o trabalho executado pelo sistema, quando instruído.

O microprocessador é, pois, o "cérebro do sistema". Ele contém todos os circuitos lógicos necessários para executar, com a ajuda de um programa monitor, uma lista de instruções introduzidas pelo teclado segundo um código de linguagem por ele *reconhecido.*

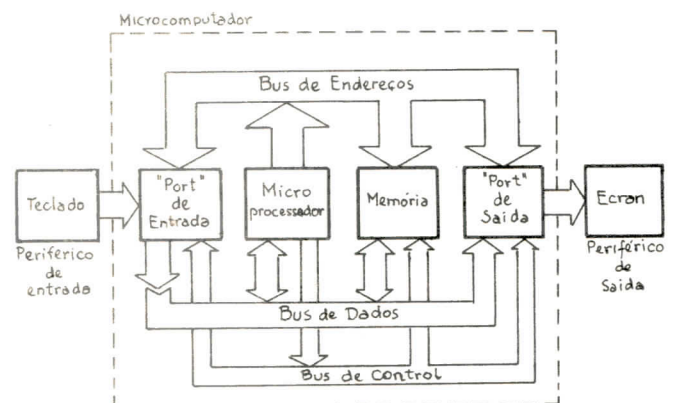



Figura 1 — Esquema básico de um microprocessador

Os blocos que pertencem ao microprocessador (zona rodeada pelo tracejado) estão interligados por meio de 3 "buses". Um "bus" é um grupo de fios que interligam os diferentes circuitos do sistema, em paralelo. O microprocessador usa o "bus de endereços" para seleccionar as posições (ou células) de memória, ou os "ports" de entrada ou de saída. Os endereços são como os números de telefone, que identificam as localizações aonde se quer colocar ou obter uma informação.

* Conforme já referido em n.º anterior, esta rubrica passa a condensar duas que se publicavam em separado — INTRODUÇÃO À LINGUAGEM MÁQUINA ZX81 e ESPAÇO SPECTRUM

Ele selecciona uma localização através do "bus de endereços" e transfere os dados através do "bus de dados". A informação pode viajar do microprocessador para a memória ou para o "port" de saída, e da memória ou "port" de entrada para o microprocessador.
O terceiro "bus" é chamado o "bus de control". Através dele circulam dois tipos de sinais:

— Os utilizados pelo microprocessador para *notificar* a memória ou os dispositivos de entrada/ /saída de que ele está pronto para realizar uma transferência de dados;
— Os utilizados pela memória ou pelos dispositivos de entrada/saída para fazerem *pedidos* especiais ao microprocessador.

Como já sabemos, o elemento mais simples usado na informação binária é o *bit* (0 ou 1). No entanto, os microprocessadores quase nunca manipulam a informação bit a bit. Eles processam em simultâneo grupos de bits chamados *palavras*.
A maioria dos microprocessadores actualmente existentes no mercado (entre eles o do ZX81 e do Spectrum) trabalham com palavras de 8 bits. Para microprocessadores deste tipo, byte e *palavra* são usados indiferentes com o mesmo significado. No entanto, saliente-se que *palavra* pode usar-se para designar um grupo de 16 ou mais bits noutros tipos de computadores.

1.3 — Programação

Como já foi dito, para obrigar um microprocessadora realizar uma dada tarefa, é necessário dispor de uma lista de instruções — cujo código ele entenda — que fiquem devidamente armazenadas na memória do sistema. Ele começará por ler a primeira instrução disponível, determina o seu significado, executa o trabalho correspondente... Depois vai ler a segunda, e assim sucessivamente.
Como a programação é o principal tema deste texto, vamos deixar os detalhes para mais tarde.

1.4 — Buffers 3-State (Tri-State)

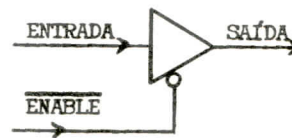
Todos os elementos dum microcomputador trocam informações com o microprocessador, através do mesmo conjunto de fios (o "bus" de dados).
O microprocessador selecciona um único elemento para colocar dados no *bus* e desliga os outros. Este tipo de funcionamento é conseguido, pelo facto de todos os elementos terem saídas *"three-state"* (3 estados).

ENABLE	ENTRADA	SAÍDA
Ø	Ø	Ø
Ø	1	1
1	Ø	-
1	1	-

Figura 2 — "Buffer 3 — State"

A figura 2 mostra o símbolo e a tabela de verdade do "Buffer 3-State", também chamado "driver 3-State".
O buffer tem uma entrada Enable* para além das habituais entrada e saída: quando a entrada Enable está a "0", o buffer funciona normalmente, transferindo o valor de entrada para a saída; Quando a entrada Enable está a "1", a saída do buffer comporta-se como se estivesse *no ar* ou desligada.
Os "drivers 3-State" são muito importantes porque permitem que muitos circuitos partilhem a mesma linha.

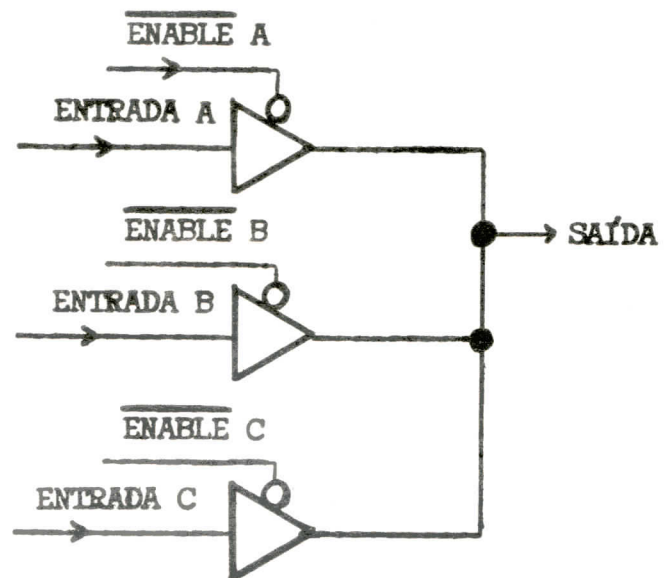



Figura 3 — Vários dispositivos partilhando a mesma linha de saída

O circuito assim formado permite que qualquer um dos elementos "3-State" possa colocar um sinal na saída. Um só Enable pode ser activado de cada vez, aparecendo na saída a respectiva entrada do driver a que esse Enable pertence.
Muitos circuitos, incluindo os microprocessadores e as memórias, contêm internamente "Drivers 3-State" na saída. Esses circuitos integrados terão uma entrada de control, muitas vezes designada por "Chip Select" (CS) ou "Chip Enable" (CE) que controla os referidos "drivers" de saída.

* O traço por cima significa que a entrada é INVERSE

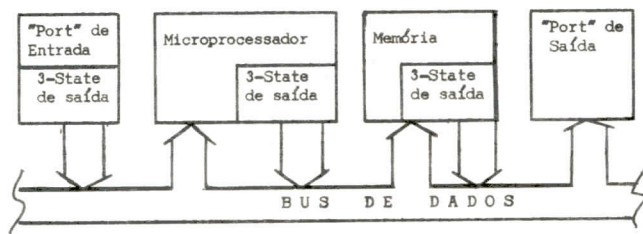



Figura 4 — Os DRIVERS 3-STATE num microcomputador

Todos os circuitos que enviam Dados para o *bus de dados* têm "drivers 3-State" nas respectivas saídas. O microprocessador *gera sinais de control* — enviados através do Bus de control — para *capacitar (pôr Enable a 0)* o circuito 3-State, do qual quer receber dados.

1.5 — O microprocessador

Quer o ZX81, quer o Spectrum, utilizam o microprocessador Z80 A da Zilog.

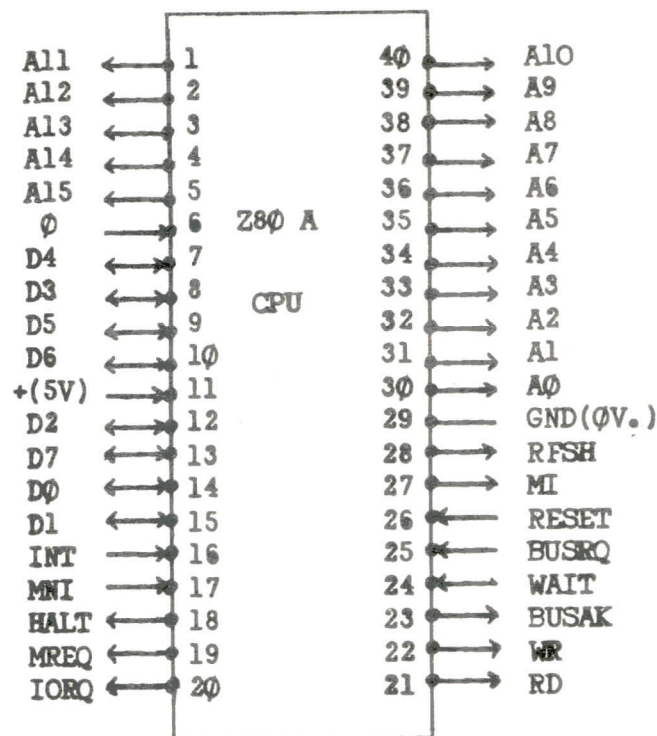



Figura 5 — Microprocessador Z80 A

Os terminais (A0 a A15) iniciam as 16 linhas de barramento (BUS) de endereços.

Os terminais (D0 a D7) iniciam as 8 linhas de barramento de dados.

Os terminais (+5 e GND) alimentam o microprocessador.

O terminal 6 é a entrada do *relógio*, que permite colocar o oscilador interno, à frequência dos impulsos de sincronismo gerados exteriormente.

Os restantes terminais servem para entrada ou saída de sinais de controlo, como se segue:

MI — Identifica o período de extracção do ciclo máquina em curso.

INT e NMI — São dois sinais de pedido de interrupção, com uma prioridade mais alta para o sinal NMI, que não pode ser desactivado por uma instrução "disable interrupts".

HALT — Toma o valor lógico 0, após a execução duma instrução HALT.

MREQ — Identifica qualquer operação de acesso à memória, em curso. É um sinal "3-STATE".

IORQ — Identifica qualquer operação IN/OUT em curso.

RFSH — Sinal utilizado para refrescar as memórias dinâmicas.

RESET — Quando se faz reset = 0, os conteúdos dos registos do microprocessador tomam o valor lógico 0, os pedidos de interrupção via INT são desactivados e todos os sinais de controlo 3-State são colocados no estado de alta impedância.

BUSRQ E BUSAK — São utilizados na técnica (DMA) de acesso directo à memória.

Mais detalhes sobre o Z80 (A), serão dados quando falarmos dos seus registos.

1.6 — Memórias

Os sistemas com microprocessadores usam normalmente memórias em circuito integrado para armazenar os programas e os dados.

Muitos bits podem ser armazenados num único CI. É normal dispormos de circuitos com 65.000 bits por "Chip". Uma memória com essa capacidade armazena cerca de 8.000 letras ou algarismos, numa placa de silício com 1 cm².

A célula mais simples de memória é um *flip-flop* que armazena 1 bit de informação.

A mais simples células integrada contém 8 flip-flops. A tecnologia LSI tornou possível a colocação de milhares dessas células, num único circuito integrado.

Claro que com milhares de flip-flops, não pode haver um pino de saída para cada um deles, nem mesmo para cada célula integrada.

A solução está no uso de entradas de endereço para seleccionar a localização da célula de memória que interessa.

O endereço é descodificado na "Chip" e a célula de memória seleccionada é ligada às entradas ou saídas de dados.

A figura 6 mostra-nos o diagrama bloco de uma célula integrada (8 flip-flops).

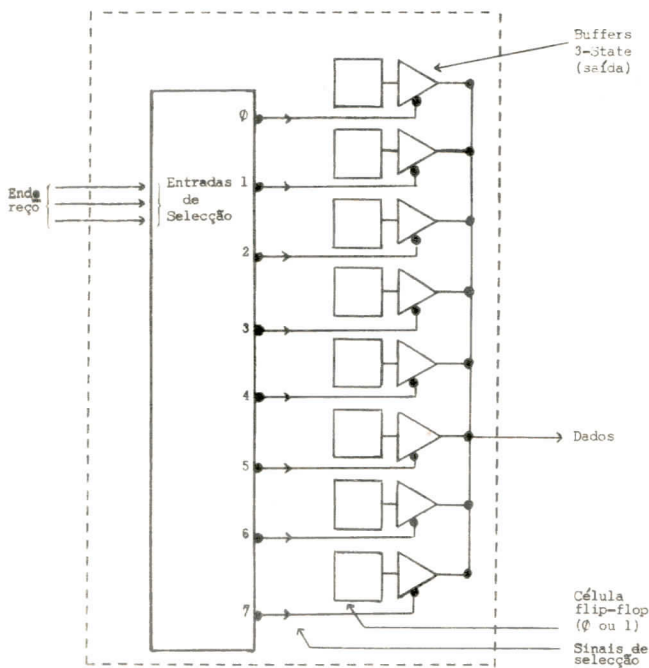



Figura 6 — Diagrama — Bloco de uma célula integrada (8 flip-flops)

A figura apenas mostra em pormenor os circuitos de saída de dados, para não complicar o diagrama. O descodificador converte as entradas (binárias) de endereços saídas separadas, uma para cada uma das possíveis configurações de endereço. Estes sinais controlam os *3 State* que comutam a saída de cada célula (flip-flop). O dado contido na célula endereçada é assim colocado na linha de saída. Por este processo, milhares destas pequenas células podem estar ligadas ao mesmo *pino de saída.*

Como o comando é *sequencial*, somente uma célula de cada vez deposita o seu sinal, no barramento de saída.

O número de posições endereçaveis (células) de uma memória depende do número de entradas de endereços que ela possui. Com uma linha de endereço, apenas se pode seleccionar uma de duas posições: a célula de endereço 0 e a de endereço 1. Com duas linhas de endereço pode ser seleccionada: endereços 00, 01, 10, 11, etc.

1.7 — RAMS e ROMS

Os circuitos integrados *memória* usados com os microprocessadores, pertencem a duas grandes categorias: ROMS E RAMS.

Uma *ROM* (Read Only memory) é uma memória que só pode ser *lida.*

Os dados são colocados dentro dela na faze de fabrico, ou ainda em certos casos, através da utilização de processos especiais imediatamente antes da sua colocação no circuito. Um programa gravado numa ROM é muitas vezes designado na literatura como "firnware".

Uma Ram (Random Acess Memory) — memória de acesso directo —, é uma memória na qual os dados podem ser escritos (armazenados) e depois lidos (recuperados). Uma característica importante das RAMS é a sua *volatilidade:* elas perdem toda a informação que contêm quando a alimentação voltar, os dados que ficam armazenados não têm qualquer significado.

A figura 7, representa uma ROM contendo 2.048 palavras de 8 bits cada, ou 16384 bits. A letra K é utilizada para designar (2^{10}) 1.024 bits. Assim, esta memória tem 2 K bytes e, como cada uma das suas células contém 8 bits, trata-se de uma ROM de 2 K x 8.

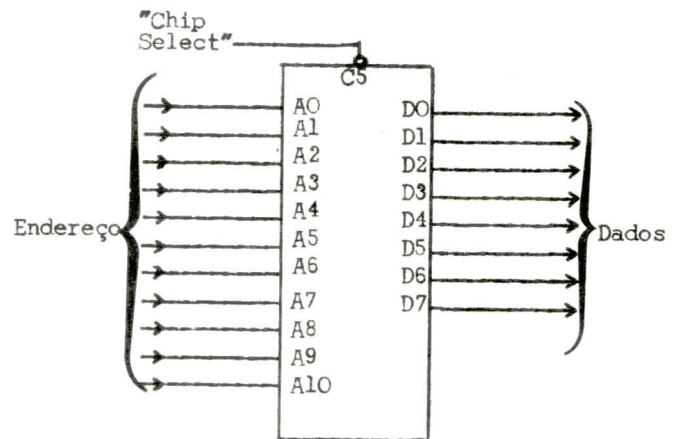



Figura 7 — ROM de 2 K x 8

Quando a entrada $\overline{CS}$ (Chip Select) for 0, o 3-State seleccionado pelo endereço, fica condutor; quando $\overline{CS}$ for 1, todas as saídas estão em aberto.

A figura 8 representa uma RAM de 1 K x 8. As linhas de dados são bidireccionais, para a informação entrar ou sair da memória.

As RAMS têm uma linha de control adicional chamada $\overline{WRITE}$.

Para armazenar dados na RAM, selecciona-se em primeiro lugar um endereço, coloca-se a informação no barramento de dados e $\overline{WRITE}$ a 0. Logo que o endereço seja introduzido, o *Chip Select* é activado e os dados ficam armazenados.

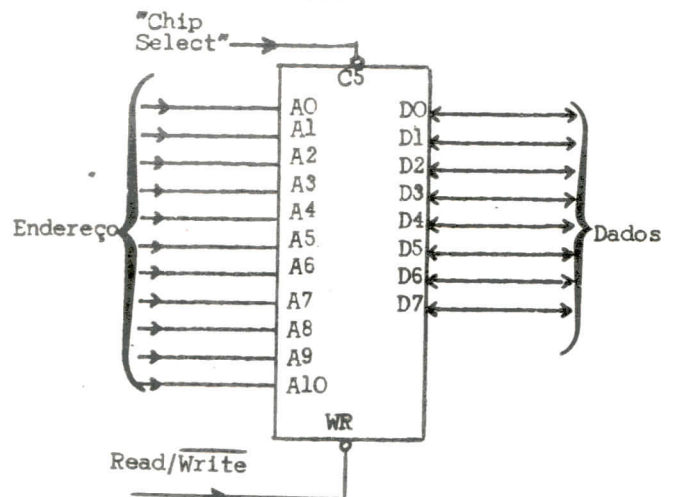



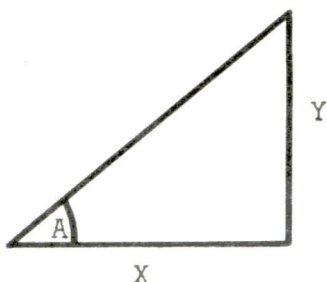
Figura 8 — RAM de 1 K x 8

A linha de $\overline{WRITE}$ determina a direcção do movimento dos dados (para dentro ou para for da memória). Esta linha é normalmente chamada RD/$\overline{WR}$ ou (R/$\overline{W}$). Esta notação indica que, se a entrada estiver a "1", é realizada uma leitura da memória e se for "0", uma escrita (ou armazenamento).

(Cont. no próximo número)

ENCICLOPÉDIA BASIC

Autor: ALEXANDRE SOUSA **MARÇO/84** **(Continuação)**

FUNÇÃO... ATN (ARCTANGENTE) ...ATAN...ATND...ATNG...ARCTAN

Esta função ATN (n) calcula o arco Tangente correspondente ao valor da razão (n). Este valor é obtido em RADIANOS... Um radiano equivale a 57 graus aproximadamente.

ARCTANGENTE (ATN) é definido como o ângulo (A) requerido para uma certa razão entre o comprimento do lado oposto ao ângulo (Y) e o comprimento do lado adjacente (X). ATN significa literalmente "O Arco (ângulo) cuja Tangente é =".

A = ATN (Y/X)

O oposto a ATN é Tangente (TAN). A Tangente do ângulo é = à razão entre o comprimento do lado oposto e o comprimento do lado adjacente.

TAN (A) = Y/X

PROGRAMA TESTE

```
10 REM "ATN" TESTE
20 PRINT "ENTRADA DO VALOR DA
   TANGENTE..."
30 INPUT N
40 LET A = ATN(N)
50 PRINT "O ANGULO CUJA TANGENTE E = ";N;"
   E IGUAL A ";A;" RADIANOS"
RUN...
ENTRADA DO VALOR DA TANGENTE...
2
O ÂNGULO CUJA TANGENTE E = 2 E IGUAL A
1.1071487 RADIANOS
```

Alguns computadores calculam o ângulo em graus ou em grados (100 grados = 90 graus). Estes computadores usam a função ATND para graus e ATNG para grados.

O programa teste que usámos anteriormente, deveria apresentar os seguintes valores (SE O SEU COMPUTADOR POSSUIR AS FUNÇÕES ATND e ATNG):

63.43495 graus e 70.4833 grados.

PARA CONVERTER VALORES QUE ESTÃO EM RADIANOS PARA GRAUS, multiplique o ângulo (em radianos) por 57.29578.

EXEMPLO:

LET D = ATN(A)X57.29578

PARA CONVERTER VALORES QUE ESTÃO EM GRAUS EM RADIANOS, multiplicar o ângulo (em graus) por .0174533.

EXEMPLO:

LET R = A (ângulo expresso em graus) * .0174533

IMPORTANTE:

MUITOS COMPUTADORES POSSUEM SOMENTE ATN COMO FUNÇÃO TRIGONOMÉTRICA INVERSA E PODEM NÃO TER ARCOS OU ARCSIN. NESSE CASO APENAS ATN PERMITIRÁ CALCULAR ÂNGULOS.

Se ATN é usada, então a Tangente é conhecida ou pode ser facilmente calculada. As fórmulas seguintes, permitem-lhe converter qualquer razão em Tangente e a partir desse ponto calcular o ângulo com ATN.

$$TAN = 1/COT \qquad TAN = \sqrt{\frac{1-\cos^2}{\cos^2}} \qquad TAN = \frac{1}{\sqrt{\frac{1-\sin^2}{\sin^2}}}$$

$$TAN = \frac{1}{\sqrt{CSC^2 - 1}} \qquad TAN = \sqrt{SEC^2 - 1}$$

ESTAS FÓRMULAS QUE FAZEM USO DAS RELAÇÕES TRIGONOMÉTRICAS PERMITEM CALCULAR FUNÇÕES INVERSAS. Por exemplo, para calcular A = Arcsec(x), use:

A = ATN (SQR(X*X—1))

AS FÓRMULAS PARA CADA UMA DAS FUNÇÕES INVERSAS, CODIFICADAS EM BASIC SÃO:

ARCCOS (X) = 1.5708 — 2*ATN (X/(1 + SQR(1—X*X)))
ARCCOT (X) = ATN (1/X)
ARCCSC (X) = ATN (1/SQR(X*X—1)
ARCSEC (X) = ATN (SQR (X*X—1))
ARCSIN (X) = 2*ATN(X/(1 + SQR(1—X*X)))

(Cont. no próximo número)

ALUNAGEM

ZX81 16K

Autor: FERNANDO PRECES
Sacavém

```
   1 REM "7"
   2 REM ** LUNAR LANDER E.E.TOZ
ER    2/2/81
   4 SLOW
   5 LET L=100
  10 POKE 16418,0
  12 PRINT "**ALUNAGEM**"
  13 PRINT "  --------",,,
  15 PRINT "A SUA  NAVE  ESTA  E
M  ORBITA  A"
  16 PRINT ,,"VOLTA DA LUA."
  17 PRINT ,,"TEM  QUE  A  ALUNA
R  SUAVEMENTE,"
  18 PRINT ,,"UTILIZANDO OS CONT
ROLES DE FORCA"
  19 PRINT ,,"DO  MOTOR, (TECLAS
 1 A 9), E  OS"
  20 PRINT ,,"CONTROLES DE ALTIT
UDE,(P SENTIDO"
  21 PRINT ,,"DOS  PONTEIROS  DO
  RELOGIO) , E",,"(0 SENT. CONTR
ARIO DOS PONTEIROS",,"DO RELOGIO
).",
  22 PRINT ,,"O MOTOR DISPARA NA
 DIRECCAO PARA"
  23 PRINT ,,"A QUAL A CAUDA APO
NTA."
  24 GOSUB 9500
  25 CLS
  28 PRINT ,,"A MUDANCA  DE  POS
ICAO DESLIGA O"
  29 PRINT ,,"MOTOR."
  30 PRINT ,,"A VELOCIDADE ORBIT
AL E ALTA. TEM",,"QUE CONTROLA-L
A ANTES QUE  FIQUE",,"MUITO BAIX
A."
  31 PRINT ,,"A  VELOCIDADE  E
POSITIVA  PARA",,"CIMA E PARA A
DIREITA.",,,"  A POSICAO  E  DEF
ENIDA  POR 16",,"QUADRANTES,LOGO
 0  E  VERTICAL,4",,"E HORIZONTA
L(DIREITA) ETC..."
  32 PRINT ,,,"CARREGUE ""N/L""
 PARA COMECAR"
  33 LET P$=INKEY$
  34 IF P$="" THEN GOTO 33
  35 LET X$=""
  36 LET M$=""
  37 LET E$=""
  38 LET B$="                    "
  39 LET MR=1600000
  40 LET H=15000
  42 LET R=1615000
  43 POKE 16418,2
  45 REM ** LUNAR G ASSUMED CONS
T    FOR ALTS<=30 KM
  46 LET G=1.62
  47 FAST
  48 LET HV=1640
  49 LET ST=0
  50 LET VV=0
  52 LET FC=1
  54 LET MP=1000
  56 LET MF=3500
  58 LET TS=0
  60 LET A=0
  61 LET AT=0
  62 LET TH=0
  64 LET B=PI/2
  66 LET ET=0
  68 LET P=0
  70 LET Z$="[illegible]"
  72 LET PX=0
  74 LET SL=6
  76 LET SC=25
  78 LET T=0
  80 LET HD=1000
 100 CLS
 200 REM **.MAIN LOOP
 202 PAUSE P
 203 POKE 16437,255
 204 LET X$=INKEY$
 205 IF NOT E$="" THEN GOSUB 900
0
 208 REM ** TIME
 210 LET T=T+P/50
 216 LET ET=ET+T
 250 REM ** BURN OR COAST
 265 IF TH>0 AND MF>0 THEN GOSUB
 1000
 330 IF TH=0 OR MF<=0 THEN GOSUB
 1500
 340 LET T=0
 400 REM ** CRASH OR LAND
 420 IF H<=2 AND ABS (HV)<=2 AND
 VV>=-5 THEN GOTO 450
 440 IF H<=2 AND ABS (HV)<=5 AND
 VV>=-10 THEN GOSUB 6400
 450 IF H<=2 AND ABS (HV)<=2 AND
 VV>=-5 THEN GOSUB 6000
 460 IF H<0 THEN GOSUB 6600
 500 REM ** ATTITUDE CHANGE
 530 IF X$<"P" OR X$>"0" THEN GO
TO 600
 540 GOSUB 2000
 600 REM ** THROTTLE CHANGE
 610 IF X$>="0" AND X$<="9" THEN
 LET TH=VAL X$
 800 REM ** STATUS AND REPORT
 810 IF H<100 THEN GOSUB 5500
 820 IF H>=1000 THEN GOSUB 3500
 830 IF H>=100 AND H<1000 THEN G
OSUB 4500
 840 IF MF<=100 AND MF>40 THEN L
ET E$="[illegible]"
 860 IF MF=0 THEN LET E$="[illegible]"
 880 IF H<=100 AND ABS (HV)>20 T
HEN LET E$="[illegible]"
 890 IF H<=100 AND VV<-30 THEN L
ET E$="[illegible]"
 900 REM ** CYCLE
 960 IF H<=1000 THEN LET P=100
 980 IF H>1000 THEN LET P=250
 990 GOTO 200
1000 REM ** BURN
1005 REM ** ORBITAL MOTION
APPROXIMATED FOR ALT <=30 KM AND
 T<=10SEC

1010 LET MF=MF-T*TH*FC
1020 LET F=TH*2000/(MP+MF-T*TH*F
C/2)
1030 LET HD=HV*T+F*(SIN A)*T**2/
2
1035 LET OC=HD/R
1040 LET VD=VV*T+(F*(COS A)-G)*T
**2/2+HD*HD/R
1045 LET H=H+VD
1050 LET R=R+VD
1090 LET HV=HV+T*F*SIN A
1095 IF ABS VV>1 THEN LET HV=HV-
VV*OC
1100 LET VV=VV+(F*(COS A)-G)*T
1120 LET VV=VV+HV*OC
1490 RETURN
1500 REM **COAST
1510 LET HD=HV*T
1515 IF HD=0 THEN LET HD=.01
1520 LET OC=HD/R
1530 LET VD=VV*T-0.8*T**2+(HD*HD
)/R
```

```
1540 IF ABS (VV)>1 THEN LET HV=H
V-VV*OC
1550 LET VV=VV-G*T+HV*OC
1560 LET R=R+VD
1570 LET H=H+VD
1990 RETURN
2000 REM ** ATTITUDE CHANGE
2010 IF X$="P" THEN LET AT=AT+1
2020 IF X$="O" THEN LET AT=AT-1
2021 GOTO 2028
2022 LET AX=AT
2024 IF AT<0 THEN LET AX=AT+16
2026 IF AT>15 THEN LET AX=16-AT
2027 LET AT=AX
2028 LET A=AT*PI/8
2029 PRINT AT 20,0;"DIRECCAO"
2030 PRINT AT 21,2;INT AT;"  "
2032 PAUSE 15
2033 POKE 16437,255
2034 LET X$=INKEY$
2036 LET T=T+.3
2038 IF X$="P" OR X$="O" THEN GO
TO 2010
2040 LET TH=0
2490 RETURN
3000 REM ** EST ORBITAL PIC
3010 GOSUB 7500
3020 PRINT AT 21,15;0
3030 PRINT AT 16,15;5000
3040 PRINT AT 11,15;10000
3050 PRINT AT 6,15;15000
3490 RETURN
3500 REM ** UPDATE ORBITAL PIC
3510 IF NOT PX=3 THEN GOSUB 3000
3512 IF SL<1 THEN LET SL=1
3514 IF SL>20 THEN LET SL=20
3520 LET PX=3
3525 PRINT AT SL,25;"▒"
3527 UNPLOT 50,2*(21-SL)
3530 LET SL=21-H/1000
3540 IF SL<0 THEN GOTO 3600
3550 PRINT AT SL,25;"S"
3600 GOSUB 7000
3990 RETURN
4000 REM ** EST INTER PIC
4010 GOSUB 7500
4020 PRINT AT 21,15;0
4030 PRINT AT 16,15;250
4040 PRINT AT 11,15;500
4050 PRINT AT 6,15;750
4490 RETURN
4500 REM ** UPDATE INTER PIC
4510 IF NOT PX=4 THEN GOSUB 4000
4520 LET PX=4
4525 PRINT AT SL,SC;"▒"
4527 UNPLOT 2*SC,2*(21-SL)
4530 LET SL=21-H/50
4532 LET SC=18+HD/100
4534 IF SC<16 THEN LET SC=30
4536 IF SC>30 THEN LET SC=16
4540 IF SL<0 THEN GOTO 4600
4550 PRINT AT SL,SC;"S"
4600 GOSUB 7000
4990 RETURN
5000 REM ** EST LOW PIC
5010 GOSUB 7500
5020 PRINT AT 21,15;0
5030 PRINT AT 16,15;25
5040 PRINT AT 11,15;50
5050 PRINT AT 6,15;75
5490 RETURN
5500 REM ** UPDATE LOW PIC
5510 IF NOT PX=5 THEN GOSUB 5000
5520 LET PX=5
5523 PRINT AT (SL-2),SC;"▒"
5524 PRINT AT (SL-1),(SC-1);"▒▒▒
"
5525 PRINT AT SL,(SC-1);"▒▒▒"
5527 UNPLOT 2*SC,2*(21-SL)
5530 LET SL=INT (21-H/5)
5531 IF SL>20 THEN LET SL=21
5532 LET SC=18+HD/10
5534 IF SC<16 THEN LET SC=28
5536 IF SC>28 THEN LET SC=16
5540 IF SL<0 THEN GOTO 5600
5550 GOSUB 7600
5600 GOSUB 7000
5790 RETURN
5990 RETURN
6000 REM ** GOOD LANDING
6010 PRINT AT 5,16;"BOA ALUNAGEM
6015 PRINT AT 6,15;"PARABENS"
6020 GOSUB 5500
6030 PRINT AT 8,15;"CERREGUE ""N
/L"""
6040 PRINT AT 9,15;"PARA TENTAR
SUBIR"
6050 INPUT X$
6052 PRINT AT 5,16;Z$
6054 PRINT AT 6,16;Z$
6056 PRINT AT 8,16;Z$
6058 PRINT AT 9,16;Z$
6060 LET TH=9
6065 LET VV=10
6066 LET H=10
6068 LET MP=500
6070 LET P=100
6099 RETURN
6400 REM ** HARD LAND
6405 GOSUB 5500
6410 PRINT AT 5,14;"ALUNAGEM DIF
ICIL"
6420 PRINT AT 2,0;"VOCE ESTA COM
 UMA PERNA PARTIDA E TEM 6 HORAS
 DE RESERVA DE AR"
6430 PRINT AT 8,0;"A SUA CHANCE
DE SER SALVO E        PEQUENA"
6440 STOP
6490 RETURN
6500 REM ** CRASH
6600 GOSUB 5500
6605 PRINT AT 5,16;"CRASH"
6610 PRINT AT 6,18;"CRASH"
6620 PRINT AT 7,20;"CRASH"
6630 PRINT AT 8,12;"VOCE ESTA MO
RTISSIMO"
6710 STOP
6990 RETURN
7000 REM ** REPORT STATUS
7010 PRINT AT 2,0;"ALTIT."
7020 PRINT INT H;" M   "
7030 PRINT
7040 PRINT "VEL.HORIZ"
7050 PRINT INT HV;" M/SEG   "
7060 PRINT
7070 PRINT "VEL.VERT"
7080 PRINT INT VV;" M/SEC   "
7090 PRINT
7100 PRINT "COMBUST."
7110 PRINT INT MF;" KG.   "
7112 IF MF<=0 THEN PRINT AT 12,0
;"VAZIO   "
7120 PRINT
7130 PRINT "FORCA MOT."
7140 PRINT TH
7150 PRINT
7160 PRINT "TEMPO"
7177 PRINT ET;" SECS.   "
7490 RETURN
7500 REM ** SPACE
7510 CLS
7520 PRINT AT 0,3;"** A L U N A
G E M **"
7530 FOR I=1 TO 21
7540 PRINT AT I,16;Z$
7550 NEXT I
7590 RETURN
7600 REM ** PRINT LEM PIC
7602 IF H<0 AND VV<-10 THEN GOTO
 7700
7610 IF SC>30 THEN LET SC=16
7612 IF SC<16 THEN LET SC=30
```

```
7620 IF SL<2 THEN LET SL=2
7625 IF SL>21 THEN LET SL=21
7630 PRINT AT SL,(SC-1);"█"
7640 PRINT AT SL,(SC+1);"█"
7650 PRINT AT (SL-1),(SC+1);"█"
7660 PRINT AT (SL-1),(SC-1);"█"
7670 PRINT AT (SL-1),SC;"S"
7680 PRINT AT (SL-2),SC;"█"
7690 RETURN
7700 PRINT AT 21,(SC-4);"▀ ▒▒▒ ▀"
7790 RETURN
9000 REM ** EMERGENCY MESSAGE
9005 IF ET<=ST+10 THEN RETURN
9007 LET ST=ET
9010 FOR I=1 TO 5
9020 PRINT AT 21,0;E$
9030 PAUSE 25
9035 POKE 16437,255
9040 PRINT AT 21,0;B$
9050 PAUSE 25
9055 POKE 16437,255
9060 NEXT I
9070 PRINT AT 0,0
9080 LET E$=""
9490 RETURN
9500 FOR O=1 TO 300
9510 NEXT O
9520 RETURN
```

REPÚBLICA DAS BANANAS

ZX81 / SPECTRUM 16 K

In. MICRO-INFORMATIQUE, Maio/83
Adap. e Trad.: ALEXANDRE SOUSA

Este jogo consiste em governar um país, administrando as finanças a partir de indicações fornecidas ao longo do jogo. Tem um certo interesse porque põe à prova a sua coragem: conseguirá acabar vivo?

```
  10 REM ***** O PRESIDENTE ****B
  15 SLOW
  20 PRINT "ACABA DE SER ELEITO PRESIDENTE  DA REPUBLICA DO ESTADO DA        AFRI
CA CENTRAL","DE FACTO HOUVETRUQUE COM AS ELEICOES MAS NAO   HA PROBLEMA."
  21 PRINT
  22 PRINT "AQUI ESTAO ALGUMAS IDEIAS PARA O AJUDAR A FAZER EVOLUIR O PAIS, ACTU
ALMETE EM CRISE"
  23 PRINT
  24 PRINT "* CADA HABITANTE DISPENSA 100.00 POR ANO"
  25 PRINT "* A VOSSA RECEITA VEM DAS TAXAS   INDUSTRIAIS, DO TURISMO, DA      VE
NDA DE CEREAIS E DA            TRANSFORMACAO DOS CAMPOS EM      ZONAS INDUSTRIAIS.
"
  26 PRINT "* CADA ANO DIVIDE O ORCAMENTO DO  ESTADO, PELA LUTA ANTI-POLUICAO, P
ELOS HABITANTES, EDUCACAO E AGRICULTURA."
  27 GOSUB 2700
  28 PRINT "* A EDUCACAO CORRECTA DE UM HABITANTE CUSTA 10F POR ANO."

  29 PRINT
  30 LET N5=8
  31 PRINT "* UMA LUTA EFICAZ ANTI-POLUICAO   CUSTA 0,44F POR UNIDADE DE       PO
LUICAO."
  32 PRINT
  33 PRINT "* SEM VOS QUERER DESILUDIR DEVO   DIZER QUE NENHUM DOS PRESIDENTESQU
E QUE VOS PRECEDERAM NAO      TERMINARAM O SEU MANDATO."
  35 PRINT
  39 GOSUB 2700
  40 LET N4=8
  50 PRINT
  60 PRINT "BOA SORTE PARA OS VOSSOS ",N4," ANOS"
 390 RAND
 400 LET M=69000+INT (RND*2000)
 410 LET P=INT (490*20)
 420 LET L=2000
 430 LET W=0
 431 LET X2=0
 432 LET E0=0
 434 LET X1=0
 440 LET E1=0
 441 LET S9=0
 442 LET S0=0
 443 LET H5=0
 444 LET S1=0
 445 LET D1=INT (P/3)
 446 LET SU=0
 450 GOTO 457
 455 PRINT "AGORA ";N;" ANOS A TIRAR."
 457 PRINT
 459 LET L1=INT (RND*5+10)
 460 LET L0=INT (RND*10+95)
 470 PRINT
 480 PRINT "TEM"
 490 PRINT M," F DE ORCAMENTO."
 500 PRINT P;" HABITANTES."
 505 IF W=0  THEN GOTO 520
 510 PRINT W;" TRABALHADORES IMIGRANTES."
 520 PRINT L;" HECTARES."
 521 IF SU=0  THEN GOTO 548
 522 PRINT SU;" F NA SUISSA."
 523 PRINT
 524 PRINT "DESEJA RETIRAR O CAPITAL DA SUISSA ?"
 525 INPUT O$
 526 CLS
 527 IF O$(1)<>"S" THEN GOTO 548
 528 PRINT "MONTANTE A TRANSFERIR ?"
 529 INPUT SUT
 530 IF SUT>SU THEN GOTO 529
 531 LET SU=SU-SUT
 532 LET M=M+SUT
 535 CLS
 536 GOSUB 2250
 548 PRINT
 549 PRINT "ESTE ANO OS INDUSTRIAIS OFERECEM ";L0;" F POR HECTARE E AS       PLAN
TACOES CUSTAM ";L1;" F POR HECTARE."
 550 IF S1=0 THEN GOTO 570
 560 PRINT "O NIVEL DE POLUICAO ATINGIDA E  DE ",S1," UNIDADES MKSA."

 570 PRINT
 580 PRINT "QUANTOS HECTARES DESEJA VENDER  AOS INDUSTRIAIS ?"
 590 INPUT L2
 600 IF L2<0 THEN GOTO 590
 610 IF L2<L-1000 THEN GOTO 680
 620 CLS
 630 PRINT "NAO TEM MAIS DO QUE ";L-1000;" HECTARES DE TERRA DISPONIVEIS."

 640 IF X1<>0 THEN GOTO 470
 650 PRINT "OS INDUSTRIAIS NAO ESTAO         DISPOSTOS A COMPRAR A FLORESA   PELO
S CUSTOS DE ABATIMENTO."
 660 LET X1=1
 670 GOTO 470
 680 LET M=INT (M+L2*L0)
 690 LET L=INT (L-L2)
 700 LET X9=INT (44*(2000-L))
 710 LET M2=0
 720 LET M3=0
 730 LET M4=0
 740 GOSUB 2250
 750 PRINT "QUANTOS ESCUDOS DESEJA          DISTRIBUIR AOS SEUS HBITANTES ?"

 760 INPUT M1
 770 IF M1<0 OR M1>M THEN GOTO 760
 810 LET M=INT (M-M1)
 820 IF M=0 THEN GOTO 1170
 825 GOSUB 2250
 830 PRINT "QUANTOS HECTARES DESEJA CULTIVAR ?"
 840 INPUT L3
 850 IF L3<0 THEN GOTO 840
 860 IF L3<=P*2 THEN GOTO 890
 865 CLS
 870 PRINT "CADA HABITANTE NAO CULTIVA MAIS DO QUE 2 HECTARES."
 880 GOTO 950
 890 IF L3<=L-1000 THEN GOTO 920
 900 PRINT "NAO TEM MAIS DO QUE ";L-1000;"  HECTARES DE TERRA PARA CULTIVAR."

 910 GOTO 950
 920 LET M4=INT (L3*L1)
 930 IF M4<=M THEN GOTO 990
 940 GOSUB 2250
 950 PRINT "O VOSSO ORCAMENTO LIMITA-VOS A ";INT (M/L1);" HECTARE DE PLANTACOES.
"
 960 PRINT "A POPULACAO PODE TRABALHAR";P*2;" HECTARES."
 980 GOTO 830
 990 LET M=INT (M-M4)
1000 IF M=0 THEN GOTO 1170
1005 GOSUB 2250
1010 PRINT "QUANTO PRETENDE DISPENSAR PARA A EDUCACAO ?"
1020 INPUT M2
1030 IF M2<0 OR M2>M THEN GOTO 1020
1070 LET M=INT (M-M2)
1080 IF M=0 THEN GOTO 1170
1090 IF L=2000 THEN GOTO 1170
1100 GOSUB 2250
1105 PRINT "QUE QUANTIDADE PENSA DISPENSAR  PARA A LUTA CONTRA APOLUICAO ?"

1110 INPUT M3
1120 IF M3<0 OR M3>M THEN GOTO 1110
1160 LET M=INT (M-M3)
1170 CLS
1180 LET D0=0
1190 LET E2=M2/P
1200 LET D2=INT (P-M1/100)
1210 IF D2<=0 THEN GOTO 1240
1220 LET D0=D2
1230 PRINT D2," HABITANTES MORTOS DE FOME."
1240 LET D3=INT (S1/100000*(.1+.4*RND)*P)
1250 IF D3<=0 THEN GOTO 1280
1260 PRINT D3;" HABITANTES MORTOS POR EXCESSO DE POLUICAO."
1270 LET D0=D0+D3
1280 IF D0<=0 THEN GOTO 1400
1290 LET F=D0*9
1300 PRINT "E OBRIGADO A DISPENSAR ";F;" PARA OS ENTERRAR."
1310 LET M=INT (M-F)
1320 IF M>=0 THEN GOTO 1400
1330 PRINT "O BAIXO ORCAMENTO OBRIGA-O A VENDER TERRENO."
1340 LET T0=-INT (M/L0)+1
1350 LET L=L-T0
1360 IF L>=1000 THEN GOTO 1390
1370 PRINT "NAO TEM TERRENO SUFICIENTE."
1380 GOTO 2270
1390 LET M=M+T0*L0
1400 IF D0>200 THEN GOTO 2300
1410 LET P=P-D0
1420 LET D1=D1-D0
1430 IF D1<0 THEN GOTO 2410
1440 IF D2<=2 THEN GOTO 1460
1450 IF M>500 THEN GOTO 2520
1460 IF L2=0 THEN GOTO 1520
1470 LET T0=INT (L2+.2*L2*RND)
1480 IF W=0 THEN GOTO 1500
1490 LET T0=INT (T0+.1*W)
1500 PRINT T0," TRABALHADORES IMIGRANTES        CHEGARAM."
1510 LET W=W+T0
1520 LET T0=INT (((500-P)/10-D3/3-D2/5)*.75*(1+RND))
1525 IF T0=0 THEN GOTO 1600
1530 PRINT ABS T0;" HABITANTES";
1540 IF T0<M THEN GOTO 1570
1550 PRINT "VIERAM SE INSTALAR."
1560 GOTO 1590
1570 PRINT "DEIXARAM O PAIS."
1590 LET P=P+T0
1600 IF P<W THEN GOTO 2480
1610 LET T0=0
1620 IF L=2000 THEN GOTO 1680
1630 LET T0=INT (S1/100000*L3)
1640 IF T0<=L3 THEN GOTO 1660
1650 LET T0=L3
1660 IF T0=0 THEN GOTO 1680
1670 PRINT "TEM";L3;" HECTARES PLANTADOS."
1680 PRINT "COLHEU";L3-T0;" HECTARES."
1690 IF T0=0 THEN GOTO 1760
1700 IF X2>=2 THEN GOTO 1760
1710 PRINT "(A CAUSA DO AUMENTO DE          POPULACAO)"
1750 LET X2=X2+1
1760 LET T1=INT ((39+RND*20)*(1+V25*(E0+E1)/20))
1770 PRINT "O QUE IMPORTOU ",INT (T1*(L3-T0));"F."
```

```
1780 LET M=M+INT (T1*(L3-T0))
1790 LET E0=E1
1800 IF E2<=10 THEN GOTO 1820
1810 LET E2=10
1820 LET E1=E2
1830 LET T0=2000-L
1840 IF T0<2 THEN GOTO 2100
1850 IF T0<=26 THEN GOTO 1870
1860 LET T0=26
1870 LET T0=INT (T0*500*(52+.5*RND))
1880 LET T1=S1/130000
1890 IF T1<=1 THEN GOTO 1910
1900 LET T1=1
1910 LET T1=T1*T0
1930 LET M=M+INT (T0-T1)
1935 PRINT "O TURISMO";
1940 IF INT (T0-T1)<=M5 THEN GOTO 2080
1950 IF S1<=S9 THEN GOTO 2080
1960 PRINT " , COMEÇA A HAVER DIMINUIÇOES";
1970 LET T2=INT (RND*5+1)
1980 GOTO 1970+T2*20
1990 PRINT "GRANDE NUMERO DE PEIXES MORTOS.";
2000 GOTO 2080
2010 PRINT "AS AVES ESTAO A DASAPARECER,";
2020 GOTO 2080
2030 PRINT "OS RIOS PROCURADOS PELOS        TURISTAS ESTAO POLUIDOS,";

2040 GOTO 2080
2050 PRINT "O AR POLUIDO AFASTA O TURISMO,";
2060 GOTO 2080
2070 PRINT "OS HOTEIS ESTAO POLUIDOS COM O  FUMO DAS FABRICAS,";
2080 PRINT "IMPORTOU ";INT (T0-T1+;"F."
2085 LET M5=INT (T0-T1)
2090 LET S9=S1
2100 LET S2=(2000-L)*(2000-L)-M3/.44
2105 IF X9=0 THEN GOTO 2120
2110 PRINT "RENDIMENTO DAS TAXAS INDUSTRIAIS";X9;" F."
2120 IF S2>=0 THEN GOTO 2170
2130 LET S1=INT (S1+S2/2)
2140 IF S1>=S0 THEN GOTO 2160
2150 LET S1=S0
2160 GOTO 2190
2170 LET S1=INT (S1+S2)
2180 LET S0=INT (S0+S2/10)
2190 LET N5=N5-1
2200 GOSUB 2700
2210 IF N5=0 THEN GOTO 2570
2240 GOTO 455
2250 CLS
2255 PRINT "POSSUI";M;"F."
2257 PRINT
2260 RETURN
2270 PRINT ".........."
2280 STOP
2300 PRINT
2310 PRINT D0;"MORRERAM ESTE ANO MUITAS PESSOAS DEVIDO AOS SEUS ERROS."
2320 PRINT ".."
2330 LET T0=INT (RND*3+1)
2340 GOTO 2330+20*T0
2350 PRINT "."
2360 STOP
2370 PRINT "..."
2380 STOP
2390 PRINT "VOCE E O INIMIGO PUBLICO N. 1..."
2400 STOP
2410 PRINT
2420 PRINT "MAIS DE 1/3 DA POPULACAO MORREU DEPOIS DA SUA ELEICAO"

2430 PRINT
2440 PRINT "OS SOBREVIVENTES APENAS DESEJAM A SUA MORTE"
2450 STOP
2480 PRINT
2490 PRINT "OS TRABALHADORES NA SUA MAIORIA IMIGRARAM."
2500 PRINT "SENTEM-SE REVOLTADOS."
2510 GOTO 2270
2520 PRINT
2530 PRINT "A FALTA DE DINHEIRO VAI IMPLICAR MAIS PESSOAS MORTAS NA MISERIA."
2540 PRINT
2550 PRINT "A POPULACAO TOMA DE ASSALTO O VOSSO PALACIO OBRIGANDO-O AO SUICIDIO.
.."
2560 STOP
2570 PRINT
2580 PRINT "FELICITACOES"
2590 PRINT
2600 PRINT "CUMPRIU CORRECTAMENTE O VOSSO   MANDATO DE";N4," ANNOS."

2610 PRINT
2611 PRINT "DESEJA TRANSFERIR O CAPITAL PARA A SUISSA ?"
2612 INPUT Y$
2613 CLS
2614 IF Y$(1)<>"S" THEN GOTO 2625
2615 GOSUB 2250
2616 PRINT
2617 PRINT "MONTANTE A TRANSFERIR ?"
2618 INPUT SUT
2619 IF SUT>M THEN GOTO 2618
2620 LET SU=SU+SUT
2621 LET M=M-SUT
2625 CLS
2629 PRINT "DESEJA SE APRESENTAR AS NOVAS ELEICOES ?"
2630 INPUT Y$
2640 LET N5=N4
2650 IF Y$(1)<>"S" THEN GOTO 280
2660 GOTO 440
2700 PRINT TAB (10);"(PRESSIONE NEW LINE)"
2710 INPUT Y$
2720 CLS
2730 RETURN
2800 CLS
2810 IF P*67>M THEN GOTO 2860
2820 PRINT "PODE PASSAR UNS DIAS NAS BAHAMAS";
2830 IF SU=0 THEN STOP
2840 PRINT "COM";SU;"F."
2850 STOP
2860 PRINT "O VOSSO SUCESSOR NAO APRECIOU O ESTADO DAS FINANCAS."

2870 PRINT "VAI ESPERA-LO ANTES DE ABANDONAR O PAIS."
2880 PRINT "E TORTURADO E E RETIRADA A SUA CONTA DA SUISSA."
2890 PRINT "O SEU CADAVER E ATIRADO AOS CROCODILOS..."
2900 PRINT
2910 PRINT "MORAL: O CRIME NAO COMPENSA."
2920 STOP
```

NOTA: Para o Spectrum, retirar todas as instruções SLOW e FAST

PACIÊNCIA

ZX81 / SPECTRUM / 16 K

In. L'ORDINATEUR INDIVIDUEL, n.º 57 Março/84
Adap. e Trad.: J. MAGALHÃES

Se possui interesse e um pouco de paciência, aqui tem um programa com um jogo de cartas que lhe permite uma interessante utilização dos caracteres gráficos.

O programa possibilita-lhe apenas um único tipo de "paciência". Dispõe de 32 cartas que são baralhadas e apresentadas da seguinte forma: 5 linhas de cartas com a face escondida; a primeira de 5 cartas; a segunda de 4; a terceira de 3, etc.... Apenas é visível a última carta de cada linha. O baralho contém as 17 cartas restantes.

O objectivo do jogo é obter 4 colunas de 8 cartas com as seguintes características:

— Cada coluna terá de começar por um REI e por valores decrescentes a terminar num ÁS.
— As cores devem ser alternadas.

Para isso terá de introduzir as cartas do baralho no jogo, não podendo introduzir nenhuma carta que não seja a 1.ª do baralho e devendo esta corresponder à regra de valores decrescentes e de cores alternadas, como foi dito. No entanto pode trocar as cartas de uma coluna para outra, mas nesse caso todas as cartas da coluna de *saída* (cuja face é vísivel) são transferidas para a coluna de chegada. As cartas escondidas são viradas automaticamente.

Os movimentos são obtidos através dos seguintes comandos:

P — Rotação das cartas do baralho em grupos de 3.

In — Introduz a 1.ª carta do baralho na coluna n.

nn — Troca as cartas das colunas de face visível **n** a **n**. Indicar n.º da coluna de onde sai a carta e n.º da coluna para onde vai (se puder fazer a transferência).

An — Permite transferir o ÁS da coluna n (se existir), o que é bastante útil quando está bloqueada e especialmente nos casos em que existe uma carta debaixo de um ÁS.

B — Nova partida quando fica definitivamente bloqueado.

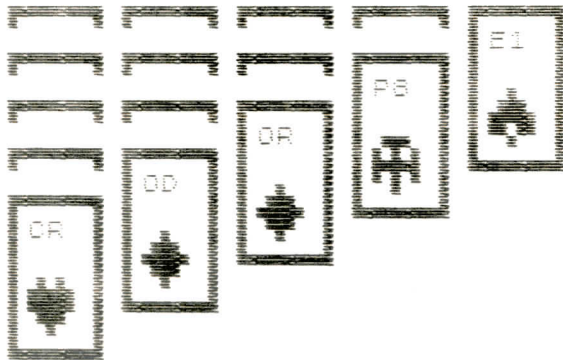
D — É equivalente (apenas para os batoteiros!) ao comando P, mas só pode ir ao baralho carta a carta.

Assim, uma coluna não pode ter mais de 8 cartas não pode incluir mais que um REI numa coluna vazia. Não haverá alteração possível numa coluna terminada.

O valor das cartas é dado por uma letra ou por um n.º:

R - Rei — D - Dama — V - Valete — X - Dez — 9 - Nove — 8 - Oito — 7 - Sete — 1 - Ás.

A cor é representada por uma letra e um desenho na 1.ª carta:

E - Espadas — P - Paus — C - Copas — O - Ouros.

```
  30 GO TO 50
  40 SAVE "PACIENCIA"
  50 CLS
  60 PRINT "UM MOMENTO..."
  70 REM
  80 DIM C$(5,8,2)
  90 DIM R(5)
 100 DIM N(5)
 110 FOR X=1 TO 5
 120 LET R(X)=1
 130 LET N(X)=6-X
 140 NEXT X
 150 LET H$="P1P7P8P9PXPVPDPRO1O7O8O9OXOVODO
RC1C7C8C9CXCVCDCRE1E7E8E9EXEVEDER"
 160 FOR X=1 TO 5
 170 FOR Y=1 TO 8
 180 IF Y>N(X) THEN  GO TO 270
 190 LET H=2*INT (RND*(LEN H$/2))+1
 200 LET C$(X,Y)=H$(H TO H+1)
 210 LET G$=""
 220 LET D$=""
 230 IF H>1 THEN  LET G$=H$( TO H-1)
 240 IF H+2<LEN H$ THEN  LET D$=H$(H+2 TO )
 250 LET H$=G$+D$
 260 GO TO 280
 270 LET C$(X,Y)="R"
 280 NEXT Y
 290 NEXT X
 300 LET T$=""
 310 LET H=2*INT (RND*(LEN H$/2))+1
 320 LET T$=T$+H$(H TO H+1)
 330 LET G$=""
 340 LET D$=""
 350 IF H>1 THEN  LET G$=H$( TO H-1)
 360 IF H+2<LEN H$ THEN  LET D$=H$(H+2 TO )
 370 LET H$=G$+D$
 380 IF H$<>"" THEN  GO TO 310
 390 CLS

 410 Print At 21,0;"           BARALHO"
 420 FOR Y=1 TO 5
 430 FOR X=1 TO 5
 440 IF C$(X,Y,1)="R" THEN  GO TO 550
 450 LET C=6*X-5
 460 LET L=2*Y-2
 470 PRINT AT L,C;"      "
 480 IF Y=N(X) THEN  GO TO 540
 490 FOR S=1 TO 5
 500 PRINT TAB C;"      "
 510 NEXT S
 520 PRINT TAB C;"      "
 530 GO TO 550
 540 GO SUB 1450
 550 NEXT X
 560 NEXT Y
 570 PRINT AT 21,17;T$( TO 2)
 580 IF T$="  " THEN  LET T$=""
 590 REM Comandos
 600 INPUT A$
 610 IF A$="P" THEN  GO TO 690
 620 IF A$(1)="A" THEN  GO TO 1570
 630 IF A$(1)="T" THEN  GO TO 760
 640 IF A$="D" AND T$<>"" THEN  GO TO 710
 650 IF LEN A$=2 THEN  GO TO 970
 660 IF A$="B" THEN  RUN
 670 GO TO 600
 690 IF T$="" THEN  GO TO 600
 700 IF LEN T$>6 THEN  GO TO 730
 710 LET T$=T$(3 TO )+T$( TO 2)
 720 GO TO 570
 730 LET T$=T$(7 TO )+T$( TO 6)
 740 GO TO 570
 750 REM Comando Tn
 760 IF T$="" THEN  GO TO 600
 770 LET X=VAL A$(2)
 780 LET Y=R(X)+N(X)-1
 790 IF Y=8 THEN  GO TO 600
 800 IF Y<1 THEN  LET Y=1
 810 LET U$=T$( TO 2)
 820 LET V$=C$(X,Y)
 830 GO SUB 1710
 840 IF P=0 THEN  GO TO 600
 850 IF N(X)=0 THEN  LET N(X)=1
 860 IF C$(X,1)="R " THEN  LET Y=0
 870 LET Y=Y+1
 880 LET C$(X,Y)=U$
 890 IF LEN T$>2 THEN  LET T$=T$(3 TO )
 900 IF LEN T$<2 THEN  LET T$="  "
 910 LET R(X)=R(X)+1
 920 LET C=6*X-5
 930 PRINT AT 2*Y-2,C;"      "
 940 GO SUB 1450
 950 GO TO 570
 960 REM COMANDO nn
 970 LET X1=VAL A$(1)
 980 LET X2=VAL A$(2)
 990 LET Y1=N(X1)
1000 IF Y1<1 THEN  LET Y1=1
1010 LET Y2=R(X2)+N(X2)-1
1020 IF Y2=8 THEN  GO TO 600
1030 IF Y2<1 THEN  LET Y2=1
1040 IF R(X1)+Y2>8 THEN  GO TO 600
1050 LET U$=C$(X1,Y1)
1060 LET V$=C$(X2,Y2)
1070 GO SUB 1710
1080 IF P=0 THEN  GO TO 600
1090 LET C=6*X1-5
1100 IF Y1=1 THEN  GO TO 1400
1110 PRINT AT 2*Y1-2,C;"      "
1120 PRINT TAB C;"      "
1130 PRINT TAB C;"      "
1140 PRINT TAB C;"      "
1150 PRINT TAB C;"      "
1160 FOR X=1 TO 2*R(X1)
1170 PRINT TAB C;"      "
1180 NEXT X
1190 IF C$(X2,1)="R " THEN  LET Y2=0
1200 FOR W=1 TO R(X1)
1210 LET C$(X2,Y2+W)=C$(X1,Y1+W-1)
1220 LET C$(X1,Y1+W-1)="R"
1230 LET R(X2)=R(X2)+1
1240 LET C=6*X2-5
1250 PRINT AT 2*(Y2+W)-2,C;"      "
1260 LET X=X2
1270 LET Y=Y2+W
1280 GO SUB 1450
1290 NEXT W
1300 IF N(X2)=0 THEN  LET N(X2)=1
1310 LET N(X1)=N(X1)-1
1320 LET R(X1)=(N(X1)<>0)
1330 IF N(X1)=0 THEN  GO TO 600
1340 LET X=X1
1350 LET Y=N(X1)
1360 LET C=6*X-5
1370 PRINT AT 2*Y-2,C;"      "
1380 GO SUB 1450
1390 GO TO 600
1400 FOR Y=0 to 6
1410 PRINT AT Y,C;"      "
1420 NEXT Y
1430 GO TO 1160
1440 REM
1450 PRINT TAB C;" ";C$(X,Y);"  "
1460 PRINT TAB C;"      "
1470 IF C$(X,Y)(1)="P" THEN  LET F$="                    "
1480 IF C$(X,Y)(1)="O" THEN  LET F$="                    "
1490 IF C$(X,Y)(1)="C" THEN  LET F$="                    "
1500 IF C$(X,Y)(1)="E" THEN  LET F$="                    "
```

```
1510 PRINT TAB C;" ";F$( TO 3);" "
1520 PRINT TAB C;" ";F$(4 TO 6);" "
1530 PRINT TAB C;" ";F$(7 TO );" "
1540 PRINT TAB C;"      "
1550 RETURN
1560 REM COMANDO An
1570 LET X=VAL A$(2)
1580 LET Y=R(X)+N(X)-1
1590 IF N(X)<2 THEN  GO TO 600
1600 IF C$(X,Y,2)<>"1" THEN  GO TO 600
1610 LET T$=T$+C$(X,Y)
1620 LET Y=Y-1
1630 LET C=6*X-5
1640 PRINT AT 2*Y-2,C;"      "
1650 GO SUB 1450
1660 PRINT TAB C;"      "
1670 PRINT TAB C;"      "
1680 IF Y<N(X) THEN  LET N(X)=Y
1690 GO TO 600
1700 REM
1710 LET P=0
 1720 IF (U$(1)="P" OR U$(1)="E") AND (V$(1)="C"
OR V$(1)="O") THEN  LET P=1
 1730 IF (U$(1)="C" OR U$(1)="O") AND (V$(1)="P"
OR V$(1)="E") THEN  LET P=1
 1740 IF U$(2)="R" AND V$="R " THEN  GO TO 1760
 1750 IF NOT P THEN  RETURN
 1760 LET P=0
 1770 IF U$(2)="1" AND V$(2)="7" THEN  LET P=1
 1780 IF U$(2)="7" AND V$(2)="8" THEN  LET P=1
 1790 IF U$(2)="8" AND V$(2)="9" THEN  LET P=1
 1800 IF U$(2)="9" AND V$(2)="X" THEN  LET P=1
 1810 IF U$(2)="X" AND V$(2)="V" THEN  LET P=1
 1820 IF U$(2)="V" AND V$(2)="D" THEN  LET P=1
 1830 IF U$(2)="D" AND V$(2)="R" THEN  LET P=1
 1840 IF U$(2)="R" AND V$(2)=" " THEN  LET P=1
 1850 RETURN
```

MAT MAT

SPECTRUM 16K

Autor: HUGO ASSUMPÇÃO
Lisboa

Este programa em Basic ocupa cerca de 9.5 K bytes. É um programa para miúdos, servindo para aprenderem tabuada e as contas. Não deixa fazer batota, insistindo sempre até acertar ou pára se errar mais de 5 vezes.

É composto por 3 partes e só com elas é que funciona:

1.º — "MAD. MAT." — Chama o Screen e o programa principal.

2.º — "MADMATSCREEN$ — É um Rótulo de programa, enquanto se aguarda o programa principal.

3.º — "MADMAT" — É o programa principal.

Para jogar, basta responder ao que se pede ou meter os números correspondentes ao "?".

```
   1 POKE 23659,2: PAPER 7: BORD
ER 7: INK 0: CLS : GO SUB 625
   3 LET y=0: DIM a(2): POKE 236
09,50: LET b$="----------------":
 LET m$="

              "
   5 CLS : PRINT "Ola!...Sr.?";
BEEP 1,30: PRINT "Desculpe!...Co
mo se chama ?": BEEP .5,0: INPUT
 "NOME:"; LINE k$
  10 CLS : PRINT "Como esta ";k$
'''"Zx Spectrum",";: FOR n=1 TO 1
4: PRINT " Muito prazer!"(n);: B
EEP .05,20: NEXT n
  12 PRINT AT 3,0;"EU Zx Spectru
m, vou-lhe ensinar MATEMATICA, P
rimeiro a tabuada; Depois as con
tas!"
  15 PRINT '"Pode escolher:"'''"1
-TABUADA"'"2-ADICAO"'"3-SUBTRACA
O"'"4-MULTIPLICACAO"'"5-DIVISAO"
  70 BEEP .05,0: INPUT "(1-5)";
LINE a$: IF LEN a$<>1 THEN GO TO
 70
  72 IF CODE a$<49 OR CODE a$>53
 THEN GO TO 70
  73 LET x=VAL a$: IF (y=0 AND x
<>1) THEN GO TO 100
  75 IF x=1 THEN GO TO 95
  80 BEEP .09,0: INPUT "Grau de
dificuldade(1-3)"; LINE a$: IF L
EN a$<>1 THEN GO TO 80
  85 IF CODE a$<49 OR CODE a$>51
 THEN GO TO 80
  90 LET d=VAL a$
  95 CLS : LET o=0: GO TO x*100
 100 REM Tabuada
 101 IF y=0 AND x<>1 THEN CLS :
PRINT "QUERIA"!!": BEEP 1,-12
 105 PRINT "Primeiro vai estudar
 a tabuada.": LET y=1
 107 PRINT '"Escolha tabuada de
1 a 10": BEEP .3,10: BEEP .1,20
 110 INPUT LINE a$: IF a$="10" O
R (CODE a$>=49 AND CODE a$<=57 A
ND LEN a$=1) THEN GO TO 115
 113 CLS : PRINT "Assim nao!": G
O SUB 630: PRINT "Responda-me co
mo deve ser!": GO TO 107
 115 CLS : PRINT AT 1,8;"ESTUDE
A": LET l=VAL a$: GO SUB 600
 117 GO SUB 2500
 120 PRINT AT 21,0;"Ja estudou t
udo (s/n)?": BEEP .3,36
 121 PAUSE 0: LET a$=INKEY$: IF
CODE a$<>115 AND CODE a$<>110 TH
EN GO TO 121
 125 CLS : IF a$="n" THEN GO TO
107
 126 PRINT "Quero ver se ja apre
ndeu tudo!"
 130 FOR n=1 TO 5: PRINT INK n;A
T 2,0;"QUANTOS SAO:": BEEP .05,1
0: BEEP .09,20: BEEP .2,26
 131 LET d=INT (RND*10)+1
 132 LET e=INT (RND*10)+1
 133 IF e>d THEN GO TO 131
 135 PRINT 'd;" X ";e;" = ?";
 140 INPUT LINE a$: IF a$<>STR$
(d*e) THEN PRINT INK n;AT 4,8;a$
;AT 10,0;"ERROU!...Nao e ";d*e;"
?": PRINT '"Estude melhor a tab
uada !": PAUSE 200: CLS : BEEP .
09,5: BEEP .1,-5: BEEP .5,-10: G
O TO 107
 145 PRINT CHR$ 8;d*e
 150 PRINT INK n'"CERTO": BEEP .
01,20: BEEP .05,30: BEEP .1,35:
PAUSE 50: CLS
 155 NEXT n
```

```
 160 PRINT "Muito bem!...";k$:
BEEP .05,5: BEEP .05,15: BEEP .0
5,20: PRINT "Quer fazer contas (
s/n)?"
 195 GO TO 1150
 200 REM Adicao
 201 RESTORE 2000
 205 LET d$=STR$ (INT (RND*100*1
00↑d)+1)
 210 LET e$=STR$ (INT (RND*100*1
0↑d)+1)
 215 IF VAL e$>VAL d$ THEN GO TO
 205
 280 LET t$=STR$ (VAL d$+VAL e$)
: LET le=0
 290 PRINT AT 12,15-LEN d$;d$: B
EEP .1,24: PRINT AT 13,15-LEN e$
-1;"+";e$: BEEP .1,15
 295 GO TO 480
 300 REM Subtracao
 301 RESTORE 2005
 305 LET d$=STR$ (INT (RND*100*1
00↑d)+1)
 310 LET e$=STR$ (INT (RND*100*1
0↑d)+1)
 315 IF VAL e$>VAL d$ THEN GO TO
 310
 360 LET t$=STR$ (VAL d$-VAL e$)
: LET le=0: LET f=705
 370 PRINT AT 12,15-LEN d$;d$: B
EEP .1,26: PRINT AT 13,15-LEN e$
-1;"-";e$: BEEP .1,10
 380 PRINT AT 14,15-LEN d$;b$( T
O LEN d$)
 390 GO TO 485
 400 REM Multiplicacao
 401 RESTORE 2010
 402 LET d$=STR$ (INT (RND*10*10
↑d)+1)
 403 LET e$=STR$ (INT (RND*10*10
↑(d-1))+1)
 404 IF VAL e$>VAL d$ THEN GO TO
 403
 405 LET t$=STR$ (VAL d$*VAL e$)
: LET le=LEN e$: LET f=700
 407 PRINT AT 11,15-LEN d$;d$: B
EEP .1,30: PRINT AT 12,15-(le-1);"
x";e$: BEEP .1,22
 410 PRINT AT 13,15-LEN t$;b$( T
O LEN t$)
 415 FOR n=le TO 1 STEP -1
 420 LET r$=STR$ (VAL d$*VAL e$(
n))
 425 FOR m=LEN r$ TO 1 STEP -1
 430 PRINT AT 14+le-n,15-le-1+n+
m-LEN r$;"?": BEEP .05,m+10
 435 LET l=VAL r$(m): GO SUB 100
0
 440 PRINT AT 14+le-n,15-le-1+n+
m-LEN r$;r$(m)
 445 NEXT m
 450 NEXT n
 452 PRINT AT 14+le,15-LEN t$;b$
( TO LEN t$)
 455 GO TO 485
 480 LET f=705
 483 PRINT AT 14+le,15-LEN t$;b$
( TO LEN t$)
 485 FOR n=LEN t$ TO 1 STEP -1
 488 PRINT AT 15+le,15-LEN t$+n-
1;"?": BEEP .05,15+n
 490 LET l=VAL t$(n): GO SUB 100
0
 493 PRINT AT 15+le,15-LEN t$+n-
1;t$(n)
 495 NEXT n
 498 GO TO 1100
 500 REM Divisao
 501 RESTORE 2015
 503 LET d$=STR$ (INT (RND*100*1
00↑d)+1)
 504 LET e$=STR$ (INT (RND*100*1
0↑d)+1)
 505 IF VAL e$>VAL d$ THEN GO TO
 504
 508 PRINT AT 12,9-LEN d$;d$: BE
EP .05,28: PRINT AT 12,11;e$: BE
EP .05,20
 510 PLOT 86,78: DRAW 0,-6: DRAW
 LEN d$*6,0
 515 LET f=701
 520 LET m=LEN e$: LET n=1: LET
z=n
 525 LET t$=STR$ INT (VAL d$/VAL
 e$)
 530 IF VAL d$( TO m)<VAL e$ THE
N LET m=m+1
 535 LET r$=d$( TO m)
 545 GO SUB 755
 550 LET r$=STR$ (VAL r$-VAL x$)
 555 FOR x=LEN r$ TO 1 STEP -1
 560 PRINT AT 12+n,9-LEN d$+x+m-
LEN r$-1;"?"
 565 LET l=VAL r$(x): GO SUB 100
0
 570 PRINT AT 12+n,9-LEN d$-1+x+
m-LEN r$;l: BEEP .02,m
 575 NEXT x
 576 LET g=LEN r$
 580 LET m=m+1
 585 IF m>LEN d$ THEN GO TO 1100
 587 PRINT AT 0,0;"Baixa o ";d$(
m): FOR w=0 TO 20: BEEP .02,20-z
: NEXT w
 588 LET r$=r$+d$(m)
 589 PRINT AT 12+n,9-LEN d$+m-LE
N r$+g;d$(m): BEEP .3,-5
 590 LET n=n+1
 595 GO TO 540
 600 REM Tabuada
 601 PRINT AT 1,18;"TABUADA DOS
"; FLASH 1;l
 605 FOR r=2 TO 11: PRINT AT r,1
8;l;" x ";r-1;" = ";(r-1)*l: BEE
P .05,10+r: NEXT r
 610 RETURN
 620 REM Sub.©opyright
 625 POKE 23659,0: PRINT AT 20,0
: FOR n=65516 TO 65535: PRINT CH
R$ PEEK n;: BEEP .01,20: NEXT n
 626 POKE 23659,2
 630 BEEP .7,-4: RETURN
 700 REM Sub.Erro
 701 GO SUB 600
 705 READ c$: PRINT #0;AT 1,0;c$
 710 FOR r=0 TO -15 STEP -5: BEE
P .02*ABS r,r: NEXT r
 715 PRINT INVERSE 1;AT 21,6;"TE
NTE DE NOVO"
 720 PAUSE 200
 725 FOR r=1 TO 11: PRINT AT r,1
8;m$( TO 13): BEEP .01,10-r: NEX
T r
 730 PRINT #0;AT 1,0;m$( TO 32)
 740 PRINT AT 21,6;m$( TO 13)
 750 RETURN
 755 REM [illegible]
 759 LET x$=STR$ (VAL e$*VAL t$(
z))
 760 PRINT AT 0,0;"Em ";VAL r$;"
 quantas""vezes ha ";e$;" ?": B
EEP .4,22
 765 PRINT AT 13,10+z;"?"
 768 LET l=VAL t$(z)
 770 IF l=0 THEN GO SUB 805
 775 GO SUB 1000
 780 PRINT AT 13,10+z;l: BEEP .0
2,z
 785 LET z=z+1
 787 IF l=0 AND z<=LEN t$ THEN G
O TO 900
 800 RETURN
```

```
 805 PRINT AT 2,0;"Nao pode have[illegible]": BEEP .9,-13
 810 PRINT AT 3,0;"Entao meta (0)  ": PAUSE 250
 815 PRINT AT 2,0;m$
 820 RETURN
 900 LET x$=STR$ (VAL e$*VAL t$(z)): LET m=m+1: IF m>LEN d$ THEN GO TO 1100
 901 IF z>=LEN t$ THEN GO TO 1100
 902 PRINT AT 0,0;"Baixa o ";d$(m): FOR w=0 TO 20: BEEP .02,20-z: NEXT w
 904 LET r$=r$+d$(m): LET g=LEN r$
 906 PRINT AT 12+n-1,9-LEN d$+g+m-LEN r$-1;d$(m): BEEP .03,m
 910 PRINT AT 0,0;m$: GO TO 759
1000 REM Sub.Entr.Dados
1001 PAUSE 0: LET a$=INKEY$: IF CODE a$<48 OR CODE a$>57 THEN GO TO 1001
1005 IF VAL a$=l THEN BEEP .1,-1: BEEP .4,5: PRINT AT 0,0;m$: RETURN
1009 LET o=o+1: IF o=6 THEN LET o=5: GO TO 1100
1010 GO SUB f
1020 GO TO 1000
1100 RESTORE 2020: FOR n=0 TO o: READ c$: NEXT n
1110 PAUSE 40: CLS : PRINT c$'"Errou ";o;" vezes no grau ";d
1115 GO SUB 2500
1120 PRINT 'k$;
1124 IF a(1)*2-a(2)>=d*2-o AND a(1)>=d THEN PRINT INK o+1;"..Ja fez melhor!": GO TO 1130
1126 PRINT INK o+1;"...Esta a melhorar!": LET a(1)=d: LET a(2)=o
1130 PRINT ''"Quer tentar outra vez(s/n)?": BEEP .3,36
1150 LET a$=INKEY$: IF CODE a$<>115 AND CODE a$<>110 THEN GO TO 1150
1160 IF a$="n" THEN CLS : PRINT FLASH 1;AT 10,10; BRIGHT 1;"ENTAO ADEUS": FOR n=1 TO 25: BEEP .05,35: BEEP .05,-30: NEXT n: PRINT USR 0
1200 CLS : PRINT "Quero ver isso!!.": GO SUB 630: BEEP .02,25: BEEP .1,32
1210 GO TO 15
2000 DATA "Falhou esta..!"
2001 DATA "Errou outra vez!?"
2002 DATA "Voce esta uma lastima..."
2003 DATA "Ja nem sabe usar os dedos  pff.."
2004 DATA "[illegible]"
2005 DATA " A subtracao nao e o seu forte !"
2006 DATA "Ai!...Perdeu outra tentativa"
2007 DATA "Ui!...Esta agora era de caras"
2008 DATA "De certeza que eu faria melhor !"
2009 DATA "[illegible]"
2010 DATA "Que pena...Ja nao tem o record!"
2011 DATA "Duas vezes nao esperava de si !"
2012 DATA "Epa!...Tem de estudar a tabuada!"
2013 DATA "E disse-me que sabia a TABUADA!?"
2014 DATA "Nao torno a jogar com [illegible]"
2015 DATA "QUERIA!..Dividir e mais dificil!"
2016 DATA "Ola!?...Voce esta muito fraco !"
2017 DATA "Nao e nao!!...Isso queria voce !"
2018 DATA "ENA!!.Essa foi de principiante"
2019 DATA "Que tal um ultimo es[illegible]"
2020 DATA " Que maquina!...Quase como EU!! "
2021 DATA "  Errar e proprio do HOMEM!!... "
2022 DATA "EU! Nao deixo passar nada.EIM?!."
2023 DATA "Se estudar e nao fizer batota,..talvez me ganhe um dia!."
2024 DATA "Nao me bata...A culpa e sua.Vocenao ESTUDA a TABUADA!!..."
2025 DATA "Voce assim nunca mais [illegible]"
2500 RESTORE 2590
2510 FOR n=1 TO 2: READ e
2520 FOR m=1 TO 3
2530 FOR r=1 TO e: READ a,b: BEEP a,b: NEXT r
2540 NEXT m
2550 NEXT n
2560 RETURN
2590 DATA 6,.5,0,.07,2,.1,4,.3,2,.1,0,.8,7,.5,0,.07,2,.1,4,.3,2,.1,0,.8,2,.5,2,.07,4,.1,5,.2,4,.1,2,.3,11
2600 DATA 4,.1,12,.1,11,.1,9,.1,7,.1,9,.1,7,.1,5,.1,4,.1,5,.07,4,.1,2,.1,0
```

DESCODIFICAÇÃO DO "HEADER" *

SPECTRUM 16 e 48 K

In. L'ORDINATEUR INDIVIDUEL, n.º 57 Março/84
Adap. e Trad.: J. MAGALHÃES

Como sabe, as gravações em cassete são mais seguras, mais rápidas e mais diversificadas no SPECTRUM do que no seu antecessor ZX81. Há 4 tipos de gravação possíveis:

0 — Programa BASIC + valores das variáveis
1 — Quadro numérico
2 — Quadro de caracteres
3 — Blocos de bytes

Conjuntamente com estes 4 tipos, é gravado também um preâmbulo de 17 bytes (HEADER: à cabeça). É este preâmbulo que nos propomos descodificar com a ajuda de um pequeno programa misto — BASIC + Linguagem Máquina (incluída no programa).

A primeira rotina em Código Máquina, implantada em 32000, faz a escuta da cassete e inclui os 17 bytes do cabeçalho (HEADER) a partir do endereço 32256.

O 1.º Byte permite determinar o tipo de gravação que é registada com os valores 0, 1, 2 ou 3 respectivamente correspondentes aos 4 tipos acima indicados.

Os 10 bytes seguintes referem-se às 10 letras do nome dado ao programa aquando da gravação.

A seguir, outros 2 bytes que correspondem ao comprimento do programa mais as variáveis, ao comprimento dos quadros e ao comprimento do bloco de bytes.

Seguem-se mais 2 bytes correspondentes a:

— n.º de linha onde se encontra o RUN (entrada em funcionamento) do programa após o seu carregamento.
— n.º do quadro (numérico ou alfanumérico) gravado.
— origem do endereço do bloco de bytes.

Os 2 últimos bytes correspondem, para o 1.º tipo de gravação, ao comprimento do programa BASIC, sem alusão às variáveis.

A 2.ª rotina em Código Máquina, implantada em 32016 permite a impressão das linhas do écran na impressora.

Este programa, uma vez carregado e depois de fazer RUN, permite fazer um sumário de registos sobre determinado programa.

Nos 2 quadros a seguir, referem-se as duas rotinas em Linguagem Máquina incluídas no programa.

* «Header» — bloco com o nome do programa

ESCUTA DA CASSETE

Etiqueta	Endereço	Mnemónicas	Cód. Decimais	Comentários
Nome do Programa	32000	LD IX, 32256	221, 33, 0, 126	Destino dos 17 bytes
	32004	XOR A	175	
	32005	LD DE, 17	17, 17, 0	17 bytes a ler
	32008	SCF	55	
	32009	CALL "LOAD"	205, 86, 5	Chamada de LOAD
	32012	CP D	186	Tem HEADER?
	32013	JR NZ LECT	32, 245	Se não, recomeçar LEITURA
	32015	RET	201	RETURN

CÓPIA DAS 9 LINHAS

	Endereço	Mnemónicas	Cód. Decimais	Comentários
	32016	LD B, 72	6, 72	9 vezes 8 mini linhas a copiar
	32018	DI	243	Suspender as interrupções
	32019	JP "COPY"	195, 175, 14	Chamada COPY

```
  10 CLEAR 31999
  20 DATA 221,33,0,126,175
  30 DATA 17,17,0,55,205,86,5,18
6,32,245,201,6,72,243,195,175,14
  40 FOR n=32000 TO 32021: READ
a: POKE n,a: NEXT n
  50 PRINT AT 10,5; BRIGHT 1; FL
ASH 1;"Inicie a cassete"
  60 RANDOMIZE USR 32000
  61 PRINT AT 10,5; FLASH 1; BRI
GHT 1;"Pare ";AT 14,3;"Use qualq
uer tecla": PAUSE 0
  65 CLS
  70 LET ix=32256
  80 LET type=PEEK ix
  90 PRINT INVERSE 1; BRIGHT 1;(
"Programa : " AND type=0)+("Quad
ro numerico : " AND type=1)+("Qu
adro de caracteres : " AND type=
2)+("Bloco de octetos : " AND ty
pe=3);
 100 PRINT " ";: FOR n=ix+1 TO
ix+10: PRINT CHR$ PEEK n;: NEXT
n
 110 PRINT AT 2,0; INVERSE 1; BR
IGHT 1;"Extensao "+("Programa+Va
riaveis :" AND type=0)+("Code: "
 AND type)
 120 PRINT TAB 14;" ";PEEK (ix+
11)+256*PEEK (ix+12);AT 5,0;
 130 IF type=1 OR type=2 THEN PR
INT INVERSE 1; BRIGHT 1;"Variave
l : ";: PRINT " ";CHR$ (PEEK (i
x+14)-32-64*(PEEK (ix+14)>192))+
("$" AND type=2): GO TO 160
 140 PRINT INVERSE 1; BRIGHT 1;(
"RUN linha n. : " AND type=0)+("
Origem do bloco : " AND type=3);
 150 PRINT " ";PEEK (ix+13)+256
*PEEK (ix+14)
 160 PRINT AT 7,0; INVERSE 1; BR
IGHT 1;("Comprimento do Programa
 : " AND type=0);
 170 PRINT " ";PEEK (ix+15)+256
*PEEK (ix+16)
 175 PRINT "            +++++++++
+++"
 180 RANDOMIZE USR 32016
 200 GO TO 50
```

FLASH

NEWBRAIN

Autor: PAULO CASTELO
Porto

Este programa escreve, ampliadas no ecran e em flash a uma frequência especificada, 6 frases de 10 caracteres cada (máximo do ecran).

```
1 REM   **   FLASH   **

2 REM   Paulo Castelo  -   NB8308241

10 IF TOP>=32766THEN 30
20 RESERVE 32770+TOP:REM faz TOP=32766
30 FOR n=1TO 255:CLOSE#n:NEXT n:PUT31
40 OPEN#0,0,"32":OPEN#6,6:OPEN#1,11,"n10"
60 HL=Pen(7):REM CHRROM Provisorio
70 REM ---- Ficheiro de Projeccao = 642
80 DIM a$(5):REM  Evita o  DIM a$(10)

100 PUT31,23,2:?"ENTRADA DE 6 LINHAS"
110 FOR n=1TO 19:PUT154:NEXT n
120 FOR n=0TO 5:PUT22,0,3,30:POKE641,161
130 LINPUT(">           <"+STR$(n/10+.1)+CHR$(28)+CHR$(26))a$:a$(n)=LEFT$(a$,10)
140 POKE641,129:PUT22,14,10+n:?a$(n):NEXT n:REM 641 = Tipo de Periferico [0ou4]
150 POKE641,161

160 INPUT("FREQUENCIA ? ")F:F=INT(F)
170 PUT31:POKE641,129

200 PUT23,0:ED=642:FOR l=0TO 5
210 Plotwip,a$(l)+"      "+STR$(-(l+1)):FOR n=0TO3:FOR c=0TO9:a1=PEEK(HL+n*96+c)
220 IF n=3THEN a2=0:a3=0:GOTO 240
230 a2=PEEK(HL+n*96+c+32):a3=PEEK(HL+n*96+c+64)
240 FOR m=0TO 3:b2=2^(2*m):b1=b2/2:A=(-((a1 AND b1)=b1)-2*((a1 AND b2)=b2)-4*((a
2 AND b1)=b1)-8*((a2 AND b2)=b2)-16*((a3 AND b1)=b1))*(1+2*((a3 AND b2)=b2))-159
*((a3 AND b2)=b2):IF A=0 THEN A=32
250 POKE ED,A:ED=ED+1:NEXT m:NEXT c
260 ED=ED+24:NEXT n:NEXT l:Plot wip

300 IF F=0THEN 370
310 a=PEEK(642):m=PEEK(643):n=PEEK(644)
320 POKE642,0:POKE643,0:POKE644,32
330 FOR l=0TO 1E4/F:NEXT l
340 GET#6,c:POKE642,a:POKE643,m:POKE644,n
350 IF c<>0THEN 380
360 FOR l=0TO 1E4/F:NEXT l:GOTO 320
370 GET#6,c:IF c=0THEN 370
380 PUT31:GOTO 100
```

CÁLCULO DE CUSTOS INDIRECTOS

In. BYTE, Fev./84
Adap. e Trad.: MARIA IRENE/ALEXANDRE SOUSA

Este simples método aritmético permite determinar a distribuição de custos indirectos, atingindo resultados muito mais rigorosos do que os obtidos com cálculos de aproximação tradicionais.

Funciona em qualquer computador e até numa máquina caculadora.

O utilizador pode determinar quaisquer custos transferidos entre os departamentos de uma empresa, com mais rapidez e exactidão, evitando aborrecer-se com a resolução de equações . . .

UM EXEMPLO — ANÁLISE DE CUSTOS PARA 4 DEPARTAMENTOS

Este exemplo de distribuição entre 4 departamentos demonstra como determinar as equações adequadas. Usam-se as folhas electrónicas de cálculo de linhas e colunas.

O custo total de cada departamento estabelece-se a partir dos encargos transferidos deste para todos os outros departamentos, mais os encargos do próprio departamento. Equacionando os custos globais do departamento A, teremos:

Custo (B para A) + Custo (C para A)
+ Custo (D para A) + Custo Directo (A)
= Custo total (A)

Para simplificar os cálculos seguintes, a equação pode expressar-se assim:

$$F(B,A) \times T(B) + F(C,A) \times T(C) + F(D,A) \times T(D) + DC(A) = T(A)$$

T — Representa o custo total do departamento referido no parentesis (por uma letra)

F — Representa o factor de distribuição entre 2 departamentos (o 1.° departamento é a fonte dos custos; o 2.° é o que recebe)

DC — Representa os encargos directos de 1 departamento.

$$\begin{array}{l} F(B,A) \times T(B) + F(C,A) \times T(C) + F(D,A) \times T(D) + DC(A) = T(A) \\ F(A,B) \times T(A) + F(C,B) \times T(C) + F(D,B) \times T(D) + DC(B) = T(B) \\ F(A,C) \times T(A) + F(B,C) \times T(B) + F(D,C) \times T(D) + DC(C) = T(C) \\ F(A,D) \times T(A) + F(B,D) \times T(B) + F(C,D) \times T(C) + DC(D) = T(D) \end{array}$$

FIG. 1 — Esta figura mostra as equações necessárias para calcular os custos indirectos para distribuição de 4 departamentos. Como nenhum departamento distribui custos por si próprio, deixou-se um vazio em cada linha.

Repare-se na simetria das equações, dispostas em linhas e colunas e organizadas por departamentos. Os cálculos feitos em linha correspondem aos custos absorvidos por um departamento; os cálculos realizados por colunas correspondem a transferências de um departamento para outros.

Depois de determinadas as equações, são necessários os factores de distribuição entre departamentos (F). Calculam-se separadamente e à parte do procedimento distribuição-cálculo. Os dados para determinar esses factores (F) devem procurar-se na contabilidade da firma.

Uma das maneiras para um departamento distribuir os custos por serviços, por exemplo, é tomar como base o número de pessoas em cada sector. Assim, se um sector abranger 1% do pessoal da firma e se todo o pessoal frequentar o bar, então 1% das despesas deste serão atribuídas àquele departamento. E calcular-se-à assim, para os outros departamentos. Um exemplo diferente é o das despesas de telefone.

Neste caso, devem ser distribuídas com base no nº de telefonemas e no nº de pessoas por sector.
Os factores de distribuição podem ainda basear-se em outros itens quantificáveis (p.ex. a área das salas, etc.), equivalentes a algarismos entre 0 e 1.
Depois de conhecidos todos os factores, podemos substitui-los por equações. Para este exemplo usamos uma tabela de factores e valores das despesas iniciais. Note-se que os espaços vazios foram preenchidos com um factor zero, que pode «tranquilizar» muitos computadores.

$$.00 \times T(A) + .10 \times T(B) + .15 \times T(C) + .15 \times T(D) + 8000 = T(A)$$

$$.15 \times T(A) + .00 \times T(B) + .20 \times T(C) + .10 \times T(D) + 7000 = T(B)$$

$$.00 \times T(A) + .10 \times T(B) + .00 \times T(C) + .30 \times T(D) + 9000 = T(C)$$

$$.15 \times T(A) + .15 \times T(B) + .10 \times T(C) + .00 \times T(D) + 12000 = T(D)$$

Agora, os termos ainda desconhecidos nestas equações são as despesas totais dos departamentos. O paradoxo está no facto de que elas não podem ser calculadas sem serem conhecidas. Aqui, é fundamental o processo de repetição/reiteração.
Assim, assumindo que as despesas totais, são as despesas directas, a primeira equação ficará:

$$0 + .10 \times 7000 + .15 \times 9000 + .15 \times 12000 + 8000 = T(A)$$

$$0 + 700 + 1350 + 1800 + 8000 = T(A) = 11850$$

O valor para as despesas totais do departamento A será mais exacto se estiver entre os valores 8000 e 11850.
Usando esse novo valor na segunda equação e outros valores assumidos para representar as despesas totais, teremos um novo valor para T(B) de 11775 que é significativamente melhor que o primeiro aproximado (7000).
Outras substituições e cálculos deste tipo relativamente aos 4 departamentos conduzem-nos às respostas finais. Quando terminados, os totais finais serão:

T(A) = 14 455
T(B) = 14 108
T(C) = 15 769
T(D) = 17 861

Conhecidos os custos totais para cada departamento, podemos calcular o custo final líquido, ou real, subtraindo os valores distribuídos por cada departamento a outro. Neste exemplo, o departamento A distribui \$2168 ao B e \$2168 ao D, ficando um custo líquido de \$10118. Os custos líquidos para os outros departamentos calculam-se de modo idêntico.
Quando as distribuições de e para todos os departamentos estiverem calculadas, a soma dos custos deve equivaler aos custos iniciais dos 4 departamentos, dado que não se pode perder nem ganhar dinheiro. Como o total inicial era \$36000, a soma dos custos líquidos deve ser \$36000.

PROCEDIMENTO EM FOLHAS ELECTRÓNICAS DE CÁLCULO

O quadro 2 apresenta a amostra de dados do nosso exemplo numa vulgar folha electrónica de cálculo.

```
        A         B     C          D          E         F          G
 1            DISTR. INDIRECTA DE CUSTOS METODO-CONTABILISTICO-
 2 ------------------------------------------------------------------
 3                      Ø          Ø          Ø         Ø          Ø
 4            DESPESAS TOTAIS   NA LINHA ACIMA
 5 PARA//DE --DPT A----DPT B----DPT C----DPT D----INICIAL--TOTAL--
 6 DPT A                Ø          Ø          Ø         Ø       8000      8000
 7 DPT B                Ø          Ø          Ø         Ø       7000      7000
 8 DPT C                Ø          Ø          Ø         Ø       9000      9000
 9 DPT D                Ø          Ø          Ø         Ø      12000     12000
10            DESPESAS LIQ. ABAIXO DESTA LINHA                 36000     36000
11                      Ø          Ø          Ø         Ø          Ø
12 ------------------------------------------------------------------
13            FACTORES DE DISTRIBUICAO
14                      Ø         .1        .15       .15
15                    .15          Ø         .2        .1
16                      Ø         .1          Ø        .3
17                    .15        .15         .1         Ø
```

QUADRO 2 — Apresentação típica de uma grelha de cálculo tipo VISICALC (APPLE) e que pode ser obtida com auxílio do VUCALC (utiliz. de máquinas SINCLAIR).
Nesta folha estão representados os dados p/ análise de uma companhia hipotética c/ 4 departamentos.

Na linha 3 estão as despesas totais de cada departamento e o total de todos os departamentos. As linhas 6 a 9 referem os cálculos das quantias distribuídas por cada departamento e também os encargos directos (iniciais) e os custos totais de cada departamento. A linha 11 corresponde às despesas líquidas de cada departamento e ao total dos departamentos. Os valores líquidos determinam-se subtraindo os valores distribuidos em cada coluna do total na linha 3. Como nenhum departamento pode distribuir mais do que 100% das suas despesas totais, a despesa líquida nunca pode ser negativa. Todavia, pode ser ligeiramente inferior devido ao arredondamento de valores acumulados durante os cálculos.

As linhas 14-17 contém os factores para calcular os custos a transferir entre cada par de departamentos. O programa em folhas electrónicas determina o tipo de relações circulares entre departamentos, substituindo os valores errados pelo termo "erro". Para evitar encher todo o ecran com mensagens de erro, a linha dos totais (B3-E3) estabelece o total por cada departamento a partir de zero. Assim, deve certificar-se sempre de que não há relações circulares quando estiver a armazenar os seus dados — senão obterá mensagens de erro quando recarregar as folhas electrónicas.

Antes de começar a resolver um problema específico, a linha dos totais deve ser copiada da coluna dos totais à direita (G6-G9) para a linha 3 (B3 a E3). Nos resultados obtidos, a soma das despesa líquidas não equivale ao total das despesas iniciais devido à demora das operações no computador. Contudo, ao fim de várias repetições, as despesas totais líquidas e as despesas totais iniciais serão as mesmas. De cada vez que primir uma tecla origina uma iteração completa, e todos os totais aumentarão em cada cálculo. Uma diminuição significa um erro em valores, cálculos ou fórmulas. Os resultados finais estão no quadro 3, que apresentaremos no próximo n.º, juntamente com a listagem do programa em BASIC para o exemplo referido.

(Cont. no próximo número)

O USO DO COMPUTADOR NA SALA DE AULA

Autor: ALEXANDRE SOUSA

A responsabilidade da modificação verificada, no sentido da disponibilidade do computador ao serviço do ensino, deve ser imputada principalmente ao avanço tecnológico, que permitiu reduzir os custos dos componentes do computador.

Existe a nível mundial uma pressão crescente no sentido de conseguir a introdução e a aceitação do computador na sala de aula, embora se reconheça que existem lacunas importantes na existênica de programas devidamente testados e com utilização imediata no sector da educação.

Alguns pontos devem ser considerados, quando tentamos estabelecer linhas gerais no sentido de alinhar vantagens (que existem) ao avaliar o computador como auxiliar da educação:

a) Familiarizar as crianças com a nova tecnologia, no sentido de que serão elas os cidadãos da sociedade informatizada.
b) Abordar o trabalho do computador, visto como a máquina que aparece a partir de uma extensão da electrónica.
c) Desenvolver a competencia em linguagens de programação, tal como o BASIC ou o PASCAL.
d) Levar à prática uma aprendizagem orientada e assistida, usando o computador como auxiliar do professor ou como "projector" de ideias ou de temas.
e) Desenvolver o pensamento e o raciocínio lógico na criança.

Iremos tentar desenvolver alguns pontos de abordagem desta ciência que é a INFORMÁTICA tendo em vista que apenas nos interessam PONTOS DE PARTIDA para as vossas experiências e tentativas futuras de ligação entre o COMPUTADOR e o ALUNO, sempre com a ideia de que o orientador desta ligação é o PROFESSOR.

Um primeiro ponto (que vai ser repetido com muita frequência) será o de frisar que:

— Qualquer utilizador da informática deve ser solicitado para PENSAR
— Deve saber especificar que INFORMAÇÃO pretende, a partir de uma BASE DE DADOS (fonte de conhecimentos arquivada na memória da máquina).
— tem de saber descrever a informação, a partir do mundo real.

Acreditamos, pessoalmente, que é mais importante saber PENSAR DE UMA FORMA LÓGICA E CLARA do que conhecer com perfeição as técnicas da programação.

Muitos professores são cépticos em relação ao valor educacional do computador e tecem mesmo críticas severas aos especialistas da Informática, dado o desconhecimento destes (na generalidade) de áreas fundamentais em matéria de pedagogia, o que conduz a uma certa dificuldade na criação de programas bem orientados.

Mas, este criticismo tem de ser vencido com humildade, tentando interessar os próprios profissionais da pedagogia, nas áreas da Informática, para que os programas dirigidos ao Ensino possuam qualidade pedagógica.

Uma palavra ainda para os professores que colocam reservas na aceitação do computador como seu auxiliar.

Existe um grande desconhecimento por parte dos professores (na generalidade), em todo o mundo, das potencialidades da máquina, no campo do ensino e, fundamentalmente, porque podem ver na máquina um competidor.

Esta é uma falsa questão. O erro existe porque o professor pode e deve dirigir a sua energia e o seu

tempo criativo para zonas bem determinadas e bem importantes da transmissão de conhecimentos. Deste modo deixará à máquina as tarefas repetitivas.

USAR A LÓGICA PARA A DESCRIÇÃO DE DADOS

A lógica é a base da programação dos computadores. Os programas incluem o acesso e a criação de BASES DE DADOS, as quais descrevem a Informação existente.
Por exemplo, nós podemos descrever a situação dos alunos na sala de aula, assinalando a cada um uma linha e uma coluna.

	A	B	C	D	E	F
1						
2	IVO					
3			RUI			
4						
5		ANA				
6				EVA		

RUI senta-se em C3
ANA senta-se em B5
IVO senta-se em A2
EVA senta-se em D6

COMO COLOCAR QUESTÕES A UMA BASE DE DADOS

UMA BASE DE DADOS contém informação, por exemplo, sobre as relações entre indivíduos e lugares. Podemos obter estas informações colocando questões do tipo:

NOME DO ALUNO *José Dias*
LUGAR OCUPADO? . . .

O computador pode dar as respostas possíveis a uma questão desta natureza!
Poderá responder com a designação do lugar do aluno . . . se este existe.
Poderá responder com uma mensagem do tipo 'ESTE ALUNO NÃO EXISTE' se o nome do aluno não faz parte da BASE DE DADOS.
O computador não toma em conta qualquer outro facto, que não seja a sua BASE DE DADOS. Ele deve estar apto a encontrar a informação pedida; caso contrário, a resposta será negativa.

UMA BASE DE DADOS PODE CONTER VOCABULÁRIO

ANA
ADA
EVA — NOMES DE ALUNOS
IVA

MAIOR DO QUE . . . relação

QUEM VAI SER RELACIONADO?

1 — aluno? EVA
2 — aluno? IVA

PERGUNTAS POSSÍVEIS:

a) EVA é maior do que IVA?
b) IVA é maior do que EVA?
c) EXISTE ALGUÉM MAIOR DO QUE EVA?
d) EXISTE ALGUÉM MAIOR DO QUE IVA?

O programa pode e deve estar preparado para dar respostas do tipo:

EVA NÃO É MAIOR DO QUE IVA

A BASE DE DADOS PODE CONTER RELAÇÕES GEOGRÁFICAS ENTRE LOCAIS

RIOS: LIMA . . . DOURO . . . MONDEGO . . . TEJO
LOCAIS: VIANA DO CASTELO . . . PORTO . . . COIMBRA . . . LISBOA

O aluno pode ser solicitado a relacionar o local com o rio que passa na sua área. O computador pode corrigir as respostas do aluno; indicar ao aluno o total de respostas certas e erradas, etc.

O COMPUTADOR COMO AUXILIAR NA MATEMÁTICA

Desde a verificação de resultados aritméticos do estilo:

4 x 5 = 22 . . . CERTO OU ERRADO?

passando pela resolução das equações . . . A = 4 + 5 até ao traçado gráfico de funções ou figuras geométricas . . . No campo da matemática, o computador é um auxiliar inatacável e de tal preciosidade que só a prática consegue demonstrar nestes termos tão encomiásticos.

NO ENSINO DE NOVAS LINGUAS

Existem programas, que são autênticas maravilhas, no sentido do auxílio pictórico e mesmo na audição dos sons característicos das palavras originais, neste caso com o auxílio dos sintetizadores de voz.

COMO INTERESSAR OS PRÓPRIOS ALUNOS NA CRIAÇÃO DAS SUAS BASES DE DADOS

Sob um ponto de vista mais avançado, ou seja, quando nós acreditamos que o professor já perdeu o receio de trabalhar com o computador, isto é, com o seu novo auxiliar, ele poderá entusiasmar os seus alunos de classes mais evoluídas, para que eles sejam envolvidos directamente na criação de BASES DE DADOS, de cada um deles.
Podemos sugerir aqui algumas ideias para projectos possíveis:

1 — Sistema solar — informações sobre os planetas; satélites; dimensões, etc.
2 — Desporto — organizar uma selecção de futebol — classificar os amigos e colegas, de acordo com as preferências desportivas.
3 — Escola — horário; disciplinas; salas, etc.
4 — Alunos — medida das alturas; tamanho dos sapatos, etc.
5 — Árvores — as que existem na zona da escola; cor das folhas; frutos, etc.
6 — Grupos — grupos formados pelos colegas.
7 — História — vestígios romanos em Portugal — batalhas importantes; seus contendores etc.
8 — Indústria — descrever uma indústria local.
9 — Automóveis — marcas; preços; velocidade, etc.
10 — Mobiliário — descrever os móveis da sala.
11 — Jogos — soluções de puzzles lógicos — resultados de jogos escolares.
12 — Catálogo da biblioteca.
13 — Moda — descritivo da moda actual; cores etc.
14 — Topografia — localização da escola.
15 — Lista dos presentes para o Natal.
16 — Como funciona uma bicicleta; quais as peças mais importantes.
17 — Carreiras de autocarros.

LOGO — UMA LINGUAGEM PARA O ENSINO

Uma linguagem de programação pode ser simples mas poderosa, ao mesmo tempo.
As linguagens de programação tradicionais foram criadas nos anos 60, com todas as dificuldades que as máquinas da época possuiam.
Em 1972, um grupo de investigadores do M.I.T., em conjunto com pessoas de outros centros de pesquisa, desenvolveram uma linguagem baseada nas experiências de aprendizagem.

LOGO é uma linguagem que proporciona vantagens nos seguintes campos:

a) Fornece um ambiente apropriado para a aprendizagem da matemática.
b) Promove o desenvolvimento da actividade ligada à resolução de problemas.
c) Serve de introdução às linguagens de programação.
d) Ajuda os estudantes a desenvolver o controlo sobre o computador.
e) Proporciona ambiente de trabalho nas áreas de música, artes plásticas, física, biologia e matemática.

NOVOS LIVROS — SPECTRUM

- **THE SPECTRUM GAMES COMPANION** *
  MAUNDER Bob, Linsac, England (Landry) (21 jogos de tipos variados, c/ explicações detalhadas).

- **EDUCATIONAL USES OF THE ZX SPECTRUM** *
  HARTNELL Tim, JOHNSON Christine e VALENTINE David, Sinclair Browne, London (Landry) (um livro dirigido aos pais e educadores: programas de matemática, línguas, leitura, gráficos... Introdução à Linguagem LOGO E PROLOG; Um dicionário no final).

- **THE ZX SPECTRUM EXPLORED** — **PREÇO**
  HARTNELL Tim, Sinclair Browne, Londres 1982. — 405$00

- **PROGRAMMING YOUR ZX SPECTRUM**
  HARTNELL Tim e JONES Dilwyin, Interface. — 600$00

NOTA: Os preços são feitos pela Landry. Portanto, não sujeitos a qualquer desconto.

* Preços ainda não fornecidos pela Landry.

Desejo trocar correspondência com outros sócios do CLUBE, sobre programação, aproveitamento de máquinas, etc.. Possuo um Spectrum 48K e coloco-me ao dispôr de qualquer associado que queira utilizar os meus préstimos aqui em Lisboa.

DANIEL DE OLIVEIRA
Av. Marechal Gomes da Costa, Lote 9
1800 LISBOA

NOVOS PROGRAMAS

SPECTRUM

	PREÇO
• **MAZIACS (16K)** — Versão do "Mazogs" do ZX81. Precisas de espadas, alimentos e sinais dos prisioneiros, de modo a combateres o malvado Maziacs. Óptima resolução gráfica.	400$00
• **PACMAN (16K)** — Moves-te num labirinto onde 4 fantasmas tenham apanhar-te. Mas as posições invertem-se se comeres um "superdot".	400$00
• **HUNGRY HORACE (16K)** — Tens que ser rápido para livrar Horace de problemas. Ele é constantemente perseguido pelos guardas do parque, porque lhes rouba o almoço, as flores e gera confusão tocando o alarme.	400$00
• **BLACK HOLE (16K)** — Naves inimigas estão a invadir o n/ Universo, através de um Buraco Negro. Terás que nos defender, usando 3 tipos de armas e 6 naves.	400$00
• **GALACTIC PATROL (16K)** — Possuis um campo magnético, um "phaser" e torpedos para te protegeres dos meteoros e naves inimigas.	400$00
• **GROUND ATTACK (16K)** — Uma batalha espacial em várias zonas com 3 naves que vão sendo destruídas pelos inimigos.	400$00
• **EMBASSY (16K)** — Jogo c/ labirinto em 3 dimensões. Podes escolher uma das 50 embaixadas, na qual tentarás apoderar-te dos códigos secretos. Agora tens que saber orientar-te para sair.	400$00
• **3 D TANQUES (16K)** — Ataca os tanques inimigos com os teus disparos. Há 3 níveis de dificuldade e possibilidade de 1,2 ou mais jogadores.	400$00
• **SUPERCHESS (16K)** — Todas as jogadas usuais. 3 níveis de dificuldade. Inclui uma "Analyse" que te ajuda a resolver problemas.	400$00
• **20 JOGOS (16K)** — Space Invaders. Laser. Lunar. Alien. Mastermind. Batalha naval. Maze. Space Roller. Skiing. Comando. Holocausto nuclear. Damas. Puckman. Escape. Aliens. Vinte e um. Reversi. Campo de minas. Firefox. Beltman. (Quase todos em Basic).	1 000$00
• **GANGSTERS (48K)** — Um jogo de estratégia baseado na "Lei Seca" nos E.U.A. em 1920. A tua vida corre perigo, pois pretendes "limpar" a cidade de quadrilhas rivais, através de negócios ilegais e subornando a polícia.	600$00
• **DITADOR (48K)** — És o Presidente de Ritimba, uma república das bananas. O teu objectivo é governar durante o máximo de tempo possível, de modo a depositares toda a fortuna na Suiça.	400$00
• **CONTABILIDADE (48K)** — Entrada/Saída Caixa-Razão — Factura etc.	1 000$00
• **DATA FILE (48K)** — Ficheiro de fácil utilização (em Português).	800$00
• **PAINT BOX (48K)** — Permite fazer desenhos c/côr e guardar na memória ou cassete p/utilização c/outros.	800$00
• **ALCHEMIST (48K)** — Descobrir o segredo das transformações de objectos.	400$00
• **STONKERS (48K)** — Jogo de estratégia em que o utilizador é comandante chefe. Possui div. infant. e pretende capturar os objectos.	400$00
• **THRUSTA (16K)** — Possui uma nave no Planeta Spectrum. Tem de destruir os ninhos dos Aliens que estão guardados e onde os ovos crescem rapidamente.	400$00
• **JOGO DE BRIDGE (48K)**	400$00
• **OMEGA RUN (48K)** — Sim. de voo adaptado c/armas anti-missil.	400$00
• **MAD MARTHA II (48K)** — Aventuras passadas em Espanha.	400$00
• **SCUBA DIVE (48K)** — Mergulhador que apanha pérolas das ostras escondidas em cavernas no mundo submarino.	400$00
• **LUNA CRABS (16K)** — Devido à falta de minerais preciosos vai ter de os procurar nas LUAS de Saturno.	400$00
• **MICROBOT (48K)** — O Robot conduz o camião, ou repara os cérebros de outros robots. Colabore no seu trabalho.	400$00
• **PANIC (16K)** — Está debaixo da terra e o oxigénio falta rapidamente. Vai ser confrontado c/ assaltantes e vai tentar escapar-lhes.	400$00
• **HUNCHBACK (48K)** — Quasimodo aprisionou Esmeralda no seu castelo. Tente saltar a ponte levadiça e salvar Esmeralda.	400$00
• **ANDROID 2 (48K)** — O Android é transportado para diferentes zonas onde terá de destruir cinco "millitoids" e regressar à sua cápsula de transporte.	400$00
• **OMNICALC (48K)** — Com características que o tornam preferencial em relação, p. ex., ao VUCALC, possibilita a construção de modelos de cálculo para uso de economistas, contabilistas, engenheiros, comerciantes, etc. Permite usar uma grela de cálculo até 5000 números, 99 colunas, 250 linhas. Permite usar todas as funções aritméticas e ainda AND, OR, ASS, RANDOM (RND). Intruções em Português.	1 000$00
• **SUPERCODE (16 e 48 K)** — V. página seguinte	700$00

DESCONTO DE 20% PARA SÓCIOS — VENDAS NA SEDE DO CLUBE OU PEDIDOS À COBRANÇA

SUPER-CODE

(SPECTRUM 16 e 48 K)

As 100 Rotinas, em código máquina, elaboradas pelo prog. Supercode:

```
          INDEX - PAGE 1
NO:                          ADDRESS
1  HIRES SCROLL-UP             64001
2  HIRES SCROLL-DOWN           64098
3  LORES SCROLL-UP             6190
4  HIRES SCROLL-RIGHT          64211
5  HIRES SCROLL-LEFT           [illegible]
6  LORES SCROLL-LEFT           [illegible]
7  LORES TOP 1/3 SCR-LEFT      64300
8  LORES MID 1/3 SCR-LEFT      64325
9  LORES LOW 1/3 SCR-LEFT      64350
10 LORES TOP 2/3 SCR-LEFT      64375
11 LORES LOW 2/3 SCR-LEFT      64400
12 LORES SCROLL RIGHT          64425
13 LORES TOP 1/3 SCR-RGHT      64450
14 LORES MID 1/3 SCR-RGHT      64475
15 LORES LOW 1/3 SCR-RGHT      64500
16 LORES TOP 2/3 SCR-RGHT      64525
17 LORES LOW 2/3 SCR-RGHT      64550
18 RIPPLE-SCROLL LEFT          64575
19 SHUTTER-SCROLL LEFT         64593
20 RIPPLE-SCROLL RIGHT         64611
          INDEX - PAGE 2
NO:                          ADDRESS
21 SHUTTER-SCROLL RIGHT        64629
22 LORES L-DIAG SCROLL         64275
23 LORES R-DIAG SCROLL         64428
24 SCREEN FILL                 64628
25 SCREEN STORE                64744
26 SCREEN OVERPRINT            64756
27 SCREEN EXCHANGE             64784
28 SCREEN INVERT               64809
29 CLEAR ALL                   0
30 INK CHANGE                  64858
31 PAPER CHANGE                64885
32 FLASH ON                    64914
33 FLASH OFF                   64951
34 BRIGHT ON                   64948
35 BRIGHT OFF                  64965
36 ATTRIBUTE FILL              64982
37 ATTRIBUTE SCR-UP            65026
38 ATTRIBUTE SCR-DOWN          65081
39 ATTRIBUTE SCR-LEFT          65204
40 ATTRIBUTE SCR-RIGHT         65143
          INDEX - PAGE 3
NO:                          ADDRESS
41 MEMORY AVAILABLE            64157
42 LINE RENUMBER               64706
43 UNI-NOTE SOUND-GEN          64647
44 DUAL-NOTE SOUND-GEN         64675
45 UNI-BEEP SIMULATOR          63000
46 MULTI-BEEP SIMULATOR        63010
47 OBLIQUE SCROLL-OFF          63034
48 ALL-LEFT SCROLL             63051
49 ALL-RIGHT SCROLL            63127
50 HIRES NW-DIAG SCROLL        63212
51 HIRES NE-DIAG SCROLL        63340
52 HIRES SE-DIAG SCROLL        63466
53 HIRES SW-DIAG SCROLL        63598
54 SCREEN-PRINT                63728
55 RANDOM NUMBER GENERATOR     63777
56 BLOCK MEMORY INSERT         63795
57 BLOCK LINE DELETE           63806
58 CHR$ SWOP                   63902
59 CHR$ SCRAMBLE               ROM
60 SUPER-RENUMBER              59294
          INDEX - PAGE 4
NO:                          ADDRESS
61 2 BYTE CONVERTER            ROM
62 DEC->HEX CONVERTER          60595
63 HEX->DEC CONVERTER          60713
64 REMKILL CONDENSER           60494
65 ON ERROR GOTO               60826
66 ON BREAK GOTO               60899
67 FREE-SCROLLER               ROM
68 NON-DELETABLE LINES         ROM
69 BORDER EFFECTS              60000
70 SCREEN SEARCH               60039
71 VARIABLES SEARCH/LIST       60222
72 24-LINE PRINTING            ROM
73 STAR/TORUS DRAW             ROM
74 FLASH SWITCH                60162
75 BRIGHT SWITCH               60192
76 PAINT-FILL                  59136
77 RECORD SOUND                65290
78 REPLAY SOUND                65318
79 SCIFI CHARACTER SET         57344
80 PROTECT PROGRAM             ROM
          INDEX - PAGE 5
NO:                          ADDRESS
81 BLOCK LINE COPY             61000
82 CONTRACT PROGRAM            61400
83 EXPAND PROGRAM              62087
84 EXPAND REM                  58892
85 APPEND STATEMENT            60407
86 ANALYSE PROGRAM             62404
87 TAPE HEADER READER          62533
88 LINE ADDRESS                59975
89 SCREEN GRID                 62819
90 MONOCHROME PROGRAM          62943
91 ANALYSE MEMORY              62857
92 HEX LOADER                  65350
93 AWAIT KEYPRESS              60972
94 U-CASE STRINGS              58833
95 L-CASE STRINGS              58774
96 U-CASE PROGRAM              58715
97 L-CASE PROGRAM              58656
98 CONFUSE LISTING             58263
99 UNCONFUSE LISTING           58398
100COMPRESS NUMBERS            58115
```

Sobre a Rubrica
O SEU NOME NOS SEUS PROGRAMAS

(V. N.° 16, JANEIRO, PÁG. 1)

Hugo Assumpção reflectiu e sugeriu o seguinte:

"Como é possível existir uma linha 0 que é indelével (pois não se pode editar) e, no entanto, continua a pertencer ao programa, podendo conter instruções realizáveis por RUN ou GOTO. No entanto não posso introduzir directamente a linha 0 REM, pois dá erro C NONSENSE IN BASIC. Também posso meter linhas depois de 9999. Ex.: No programa, POKE A, 39: POKE A, 16.

É sempre possível retirar o que está escrito nessas linhas? Sim! Na linha 0 não é possível apagar (só alterar) o que está escrito. Já no caso das linhas após 9999 é possível apagar.

O processo mais simples de aplicar o ©OPYRIGHT é ter um programa com p. ex.; 0 REM BAAL CORP e fazer MERGE "". Introduz automaticamente a linha 0 no programa e apaga a que lá estava — o que não é muito bom.

O processo de impedir que se veja o programa é usar esta linha antes de qualquer outra, no início:

```
0 LET A = PEEK 26613 + 256 * PEEK 23614:
POKE A, 0 : POKE A + 1, 0
```

e guardá-la com

```
SAVE "...." LINE 0
```

Isto impede que se faça BREAK, ou seja, limpa o computador.

C L U B E Z80

INSCRIÇÃO COMO ASSOCIADO

O **CLUBE Z80** está aberto a todos os utilizadores de microcomputadores.

A intenção de associar os entusiastas das micro-máquinas, é exclusivamente a de permitir:

1 — PUBLICAÇÃO DE UM JORNAL MENSAL, onde sejam publicados programas de uso geral ou específico como no caso da educação.

2 — PROMOVER TROCAS DE PROGRAMAS, e trocas de experiências; tanto no caso do Software (programação), como no caso do Hardware (electrónica).

3 — PROMOVER DESCONTOS NA AQUISIÇÃO DE PROGRAMAS.

4 — LANÇAR CURSOS DE PROGRAMAÇÃO EM BASIC — PASCAL OU OUTRAS LINGUAGENS E DIVULGAR O USO DE LINGUAGEM MÁQUINA.

NOME ....

IDADE .... COMPUTADOR TIPO ....

PROFISSÃO ....

ENDEREÇO ....

.... TELEF. ....

ASSINATURA ANUAL — Esc. 1 500$00 ☐

ASSINATURA SEMESTRAL — Esc. 750$00 ☐

CHEQUE OU VALE DO CORREIO

N.º ....

BANCO ....

DATA ..../......../........

JÁ SÓCIO ☐

NOVO SÓCIO ☐ → A partir do mês de .... (inclusive)